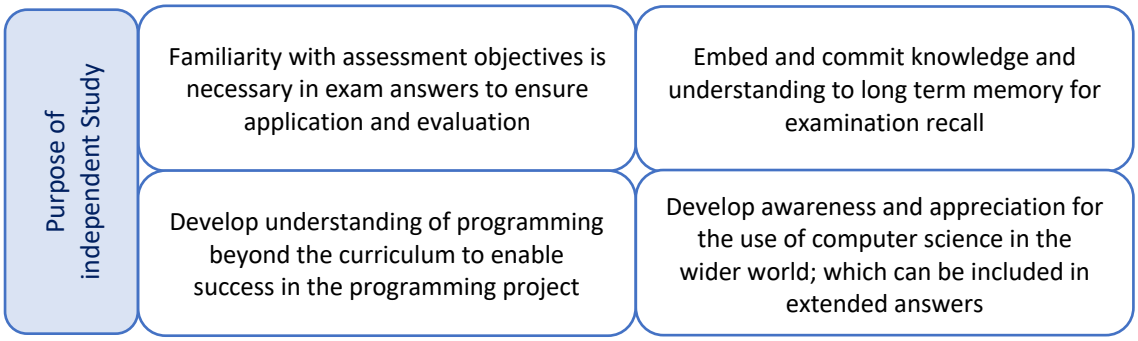
Study Skills and Support

Exam board: OCR
Course length: 2 years
How is it assessed? 2 written exams on 01 – Computer Systems and 02 – Algorithms and Programming (each worth 40%) and a programming project worth 20%.
Modules covered:
01 – Processors, Input – Output and Storage, Systems software, software development, compression, databases, networks, web technologies, data types, data structures, Boolean algebra, morals and ethics.
02 – Thinking abstractly; ahead; procedurally; logically; concurrently, programing techniques, computational methods, algorithms.



Resource	Link	Useful For…	Requirements
Course Textbook	N/A	Independent revision & study	Course textbook from the school library.
YouTube	YouTube	Knowledge booster, second voice	N/A
Past Paper Packs	N/A	Exam style question practice, independent study	Past paper pack from class teacher
Departmental resources	All stored within the Microsoft Teams Team for the group.	Accessing departmental materials and lessons	School email and password login.
Mr Fraser	www.mrfraser.org	Accessing resources and work sheets	mrfraser.org login account (free to create)
Craig n Dave	craigndave.org	Resources for topics – broken down by spec	Access is free for most content – school has a paid account
AQA Past Papers	Search 'AQA A Level Computer Science Past Papers' on Google.	Different phrasing of exam style questions.	N/A
Class Teachers	l.ravenscroft@bishopchalloner.bham.sch.uk b.ebrahim@bishopchalloner.bham.sch.uk		

OCR
Oxford Cambridge and RSA

A Level Computer Science

H446/01 Computer systems

Practice paper - Set 1

Time allowed: 2 hours 30 minutes

Do not use:
- a calculator

First name									
Last name									
Centre number						**Candidate number**				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided.
- If additional space is required, use the lined page(s) at the end of this booklet. The question number(s) must be clearly shown.
- Do **not** write in the barcodes.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **24** pages.

Answer **all** the questions.

1 A company releases a utility called RAMStore. The utility creates a virtual storage drive from an area of the computer's RAM.

(a) Describe what is meant by the term utility software.

.....

.....

.....

.....**[2]**

(b) Give **one** advantage of using RAM as storage in this way.

.....

.....**[1]**

(c) The utility periodically copies what is in the RAM drive to secondary storage, such as a hard disk. Explain why this is necessary.

.....

.....

.....

.....**[2]**

(d) It is important that enough RAM is left for the operating system to use. Describe a technique that allows operating systems to overcome a lack of available RAM.

.....

.....

.....

.....

.....

.....

.....

.....**[4]**

2 * Modern computers tend to have magnetic or solid state (flash) hard drives. Discuss which hard drive you would recommend for a keen video games player to use on their desktop PC.

.... [9]

3 **(a)** Convert the unsigned binary number 11110000 to:

(i) Denary:

.....

.....

.....**[1]**

(ii) Hexadecimal:

.....

.....

.....**[1]**

(b) An AND operation with the mask 10101010 is applied to the binary number 01010101. Show the result.

```
01010101
10101010 AND
```

[1]

(c) An OR operation with the mask 10101010 is applied to the binary number 01010101. Show the result.

```
01010101
10101010 OR
```

[1]

(d) 00001100 is shifted two places to the left.

(i) Show the result.

.....

.....

.....

.....**[1]**

(ii) Identify what arithmetic operation this shift is equivalent to.

.....

.....

.....

.....**[1]**

(e) Convert the denary number -8 to:

(i) An 8-bit sign and magnitude binary number.

.....

.....

.....

.....**[1]**

(ii) An 8-bit two's complement binary number.

.....

.....

.....

.....**[1]**

(f) A computer represents floating point binary numbers using a 6-bit mantissa and 4-bit exponent, both using two's complement.

Add the following three numbers together and give the answer in the format described.

You must show your working.

```
010100  0010

011000  0001

100010  0010
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....**[6]**

4 Below are extracts from the ASCII and EBCDIC character sets.

ASCII

Denary Value	65	66	67	68	69	70	71	72	73	74	75	76	77
Character	A	B	C	D	E	F	G	H	I	J	K	L	M
Denary Value	78	79	80	81	82	83	84	85	86	87	88	89	90
Character	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

EBCDIC

Denary Value	193	194	195	196	197	198	199	200	201	...	209	210	211	212
Character	A	B	C	D	E	F	G	H	I	...	J	K	L	M
Denary Value	213	214	215	216	217	...	226	227	228	229	230	231	232	233
Character	N	O	P	Q	R	...	S	T	U	V	W	X	Y	Z

(a) Explain, referring to ASCII and EBCDIC, what would happen if computers were to use different character sets when communicating.

.....

.....

.....

.....

.....

.....**[2]**

(b) Write a function that given the denary value of an EBCDIC uppercase letter, returns the denary value of an ASCII uppercase letter. If a value is entered that doesn't correspond to an uppercase EBCDIC letter the function should return -1

e.g.

`convert(201)` returns 73

`convert(209)` returns 74

`convert(78)` returns -1

```
function convert(ebValue)
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
endfunction
```

[5]

5 The following is a program written using the Little Man Computer instruction set.

```
start  LDA   one
       OUT
       LDA   zero
       OUT
       LDA   count
       SUB   one
       STA   count
       BRP   start
       HLT
one    DAT   1
zero   DAT   0
count  DAT   3
```

(a) Describe the difference between the `STA` and `LDA` instructions.

.....

.....

.....

.....

.....**[2]**

(b) Identify the type of memory addressing the program uses.

.....

.....**[1]**

(c) State the output this program generates.

.....

.....

.....

.....

.....

.....

.....**[3]**

(d) Explain the buses and registers used when the line `SUB one` is executed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....**[5]**

(e) Explain, giving an example, how pipelining in a CPU could speed up the execution of this program.

.....

.....

.....

.....

.....

.....**[3]**

(f) Describe **one** issue the line `BRP start` may cause for a CPU using pipelining.

.....

.....

.....

.....**[2]**

(g) Pipelining is one factor that affects the performance of a CPU. Identify **one** other factor.

.....

.....**[1]**

6 A company is writing a syntax checker to be used when writing HTML.

(a) The first thing the program does is add every tag in a piece of text to the data structure `dataStructureA`.

The string `X` is added to `dataStructureA` with the code

```
dataStructureA.add("X")
```

The string type variable `htmlCode` holds the code that is to have its tags added.

If `htmlCode` were to contain:

```
<html><head><title>My Page</title></head><body>Hello</body></html>
```

`dataStructureA` would have the following contents:

<html>	<head>	<title>	</title>	</head>	<body>	</body>	</html>

Write the code to fill `dataStructureA` with the tags in `htmlCode`.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....**[7]**

(b) Part of the program checks that the HTML tags are well formed.
Well formed HTML has tags that are nested but never overlapping.

e.g.

`<p>The cat <strong>sat</strong> on the mat.</p>` is well formed.

Whereas `<p>The cat <strong>sat</p> on the mat.</strong>` is not well formed as `p` closes before the `strong` inside it has been closed.

All comments and single tags (e.g. `img`, `br` etc) are removed from `dataStructureA`.
All attributes are removed from the within the tags.

(i) The contents of `dataStructureA` may look similar to below:

<html>	<head>	<title>	</title>	</head>	<body>	<h1>	</h1>	</body>	</html>

Tags are removed from `dataStructureA` in the same order they were added.

Identify what type of data structure `dataStructureA` is.

.....

.....**[1]**

`dataStructureB` is given a closing tag and gives the corresponding opening tag.

e.g.

```
openingTag=dataStructureB.get("</head>")

openingTag is "<head>" (courier font)
```

(ii) Identify what type of data structure `dataStructureB` is.

.....

.....**[1]**

The following code is used to check if the tags are well formed.

```
function checkTags(dataStructureA)
{
   valid=true
   //loops while code is still valid
   //and dataStructureA has tags
   while valid==true and dataStructureA.isEmpty()==false
      tag=DataStructureA.remove()
      //Next, check if closing tag
      if tag.substring(1,1)=="/" then
           if dataStructureC.remove()!=dataStructureB.get(tag)then
            valid=false
           endif
      else
           dataStructureC.add(tag)
      endif
   endwhile
   return valid
}
```

(iii) Identify what type of data structure `dataStructureC` is.

.....

..... **[1]**

(iv) Explain why `dataStructureC` is suited to checking if HTML is well formed.

.....

.....

.....

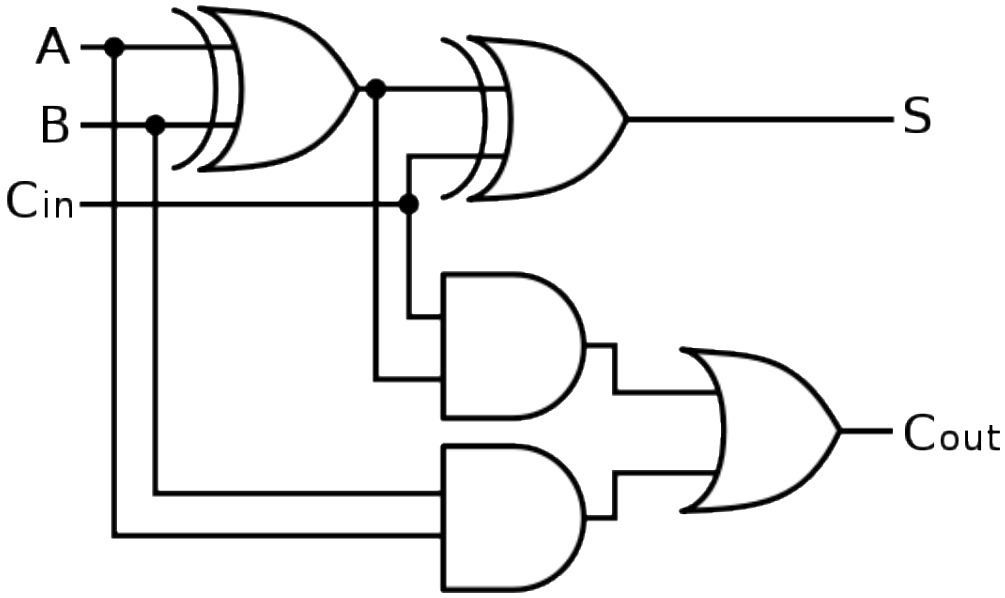
..... **[2]**

7 **(a)** An XOR gate is shown below. Complete the truth table for XOR.

A	B	Q
1	1	
1	0	
0	1	
0	0	

[2]

(b) A set of logic gates are connected as below.



(i) Complete the Truth Table below:

A	**B**	**C_{in}**	**S**	**C_{out}**
1	1	1		
1	1	0		
1	0	1		
1	0	0		
0	1	1		
0	1	0		
0	0	1		
0	0	0		

[4]

(ii) Explain what the circuit does. You should refer to A,B, C_{in}, S and C_{out} in your answer.

.....

.....

.....

.....

.....

.....

.....

.....**[4]**

(c) **(i)** Write a Boolean expression equivalent to S. **[1]**

S ≡

(ii) Write a Boolean expression equivalent to C_{out}. **[2]**

C_{out} ≡

8 A database stores information about songs on a music streaming service.

One of the tables called `Song` has the fields.

`Title`, `Artist`, `Genre`, `Length`

(a) Explain why none of these fields would be suitable as a primary key.

.....

.....

.....

.....

..... **[2]**

(b) Give **one** advantage and **one** disadvantage of indexing the field `Artist`.

Advantage .....

.....

Disadvantage .....

.....

[2]

(c) Users can build up playlists of their songs. Another table is created called `Playlist`.

Explain why a third table which we shall call `PlaylistEntry` is needed. You should use an ER diagram to illustrate your answer.

[4]

(d) A band called *RandomBits* removes their permission for their songs to be streamed.

The company removes all the songs belonging to *RandomBits* from their service.

(i) Identify the law with which the company are complying.

.....

..... **[1]**

(ii) Write an SQL statement that will remove all songs by *RandomBits* from the table `Song`.

.....

.....

..... **[2]**

(iii) When the songs have been removed, explain what must happen to the table `PlayListEntry` if the database is to retain its referential integrity. (You are not expected to write the SQL to do this).

.....

.....

..... **[1]**

9* *'Modern technology and UK laws mean privacy is dead.'*

Discuss the extent to which you agree with this statement.

....**[12]**

10 A software development company is building an operating system for a mobile phone that is in the process of being designed.

(a) Give **one** reason the phone needs an operating system.

.....

.....**[1]**

(b) Explain how the developers could use virtual machines.

.....

.....

.....

.....**[2]**

(c) One of the developers is responsible for writing the code for what happens when the CPU receives an interrupt. Outline what the code must do.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....**[6]**

(d) The developers follow the waterfall lifecycle.

(i) List **three** stages of the waterfall lifecycle.

1.....

2.....

3.....**[3]**

(ii) Justify why the waterfall lifecycle is suited to the development of the operating system.

.....

.....

.....

.....**[2]**

(iii) Give **one** disadvantage of using the waterfall lifecycle to develop the operating system.

.....

.....

.....**[1]**

(e) * The code is written using an object-oriented programming (OOP) language. Discuss the advantages and disadvantages to the team of developers of using OOP over procedural programming. You should refer to inheritance, encapsulation and polymorphism in your answer.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....**[9]**

11 A website has the following HTML code.

```
<html>
<head>
<title>My Stamp Collection - European Stamps</title>
</head>
<body>
<h1 style="font-family:Arial; color:darkGreen">United
Kingdom</h1>
<p>These are my stamps from the uk.</p>
<!-- Code A -->


<!-- Code B -->

</body>
</html>
```

the site's owner wants to add the photo UKstamps.jpg in place of the comment `<!-- Code A -->`

(a) Write the code that should go in place of the comment `<!-- Code A -->`:

.....

.....

.....

.....**[2]**

(b) Where the comment `<!-- Code B -->` is, the site's owner wants to add the text:

Find out more about UK stamps

as a link to the UK Stamp Collectors Guild website which has the URL:

http://ukstampcollectorsguild.co.uk

Write the code that should go in place of the comment `<!-- Code B -->`

.....

.....

.....

.....**[2]**

(c) The site uses styling set out as attributes in tags rather than a linked CSS file.

(i) Give **one** disadvantage of this to the site's owner.

.....

..... **[1]**

(ii) Give **one** disadvantage of this to the site's visitors.

.....

..... **[1]**

(d) The site needs a light green (web colour lightGreen) background.
Explain what change needs to be made to the current page in order to do this.

.....

.....

.....

.....

.....

..... **[3]**

(e) The site's owner notices that his site doesn't come up high in the results from a search engine that uses the PageRank algorithm. State what would affect his site's ranking.

.....

.....

.....

..... **[2]**

END OF QUESTION PAPER

ADDITIONAL ANSWER SPACE

If additional answer space is required, you should use the following lined page(s). The question number(s) must be clearly shown in the margin(s).

OCR
Oxford Cambridge and RSA

Practice Paper 1

GCE COMPUTER SCIENCE

H446/01 Computer Systems

Duration: 2 hours 30 minutes

MAXIMUM MARK 140

This document consists of 30 pages

Question			Answer	Marks	Guidance
1	a		- A piece of software… - …with one purpose… - usually to do with the upkeep/maintenance of a computer. (1 per -, max 2)	**2** **AO1.1**	
	b		Faster read/write speed than secondary storage media.	**1** **AO1.2**	
	c		- RAM is volatile meaning it loses contents when power is off … - …so must be copied to secondary storage in case of unexpected power failure (1 per -,)	**2** **AO1.2**	
	d		- Memory contents are divided into pages - Pages not needed get moved to virtual memory - Which is an area on a secondary storage device - When required the pages are moved from virtual memory back into RAM. (1 per -)	**4** **AO1.1**	
2			**Mark Band 3–High Level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of Magnetic and Flash storage. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered.	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1**	**AO1: Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: - Magnetic hard drives work by magnetic patterns being read off platters that mechanically spin at high speeds. - Flash hard drives use memory chips. These can have their contents erased and subsequently overwritten when an electrical charge is applied. - Magnetic hard drives are cheaper per GB and tend to be sold in much higher capacities than flash hard drives. - Flash hard drives tend to have much higher read/write speeds than

			There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. **Mark Band 2-Mid Level (4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of a Magnetic and Flash based storage; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-3 marks)** The candidate demonstrates a basic knowledge of Magnetic and Flash based storage with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated.	**(2)** **AO3.3** **(3)**	magnetic hard disks. - Flash hard disks have no moving parts and therefore tend to have lower power consumption and are not affected by their device moving. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: - Many games tend to incorporate a lot of media and as such a keen gamer is likely to need a lot of storage space. - Games are fast paced and often competitive. High loading speeds can be beneficial. - High performance is often important to gamers and as such will pick highest performing components. - Hybrid approaches exist which offer 'the best of both worlds' - Magnetic hard drives can be noisy (due to parts moving at high speed), this can be undesirable and distracting whilst gaming. Conversely flash drives operate silently. **AO3.3: Evaluation**

			The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear. **0 marks** No attempt to answer the question or response is not worthy of credit.		Candidates will need to consider a variety of issues in relation to the question and will make some evaluative comments about the issues and solutions they are discussing. The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: - Due to their high storage capacity magnetic hard disks are the best choice. A gamer could have many games installed at one time. Whilst performance is not quite that of flash drives , to have a similarly sized flash drive would be prohibitively expensive. A high quality magnetic drive will provide good enough performance leaving money to be spent elsewhere. As it is being installed on a desktop there is no need to worry about power consumption or issues with the computer moving. - Gamers need high performance and that includes large amounts of data being loaded quickly. The read/write speed of a solid state drive means this is the natural choice for the gamer's desktop.
3	a	i	240	**AO1.2** **1**	
		ii	F0	**AO1.2** **1**	
	b		00000000	**AO1.2** **1**	
	c		11111111	**AO1.2**	

				1	
	d	i	00110000	**AO1.2** **1**	
		ii	Multiplying by 4	**AO2. 1** **1**	
	e	i	10001000	**AO1. 1** **1**	
		ii	11111000	**AO1. 1** **1**	
	f		- Calculate the first number as: `010.100` - Calculate the second number as: `01.1000` - Calculate the third number as: `100.010` - Add the three together to get `(1)000.010` - Show carry bits. Discarding of leading one may be shown or implicit. - Normalised result is: `010000 1111` (1 per -)	**AO1.2** **6**	If candidate fails to discard the overflowing 1 they can still get marks 1-4.
4	a		- Characters from a computer using ASCII will be interpreted as different characters by a computer using EBCDIC. - Text will be incomprehensible. (1 per -)	**AO1.2** **2**	

	b		- Value between 193 and 201 returns respective ASCII value between 65 and 73
			- Value between 209 and 217 returns respective ASCII value between 74 and 82
			- Value between 226 and 233 returns respective ASCII value between 83 and 88
			- Values less than 193 and greater than 233 return -1
			- Values between 202 and 208, and 218 and 225 return -1.
			(1 per -)
				AO3.2
				
				5
					`function convert(ebValue)`
					`  if ebValue >= 193 and ebValue <= 201 then`
					`  return ebValue - 128`
					`  elseif ebValue >= 209 and ebValue <= 217 then`
					`  return ebValue - 135`
					`  elseif ebValue >= 226 and ebValue <= 233 then`
					`  return ebValue - 143`
					`  else`
					`  return -1`
					`  endif`
					`endfunction`
					
					A program that returns a value 128 less for values between 193 and 208 would receive the first mark but not the last one. (The same principle applies for points 2 and 3)

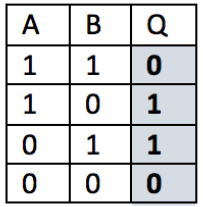
5	a		- STA store the value in the accumulator into a given memory location - LDA loads the value in a memory location into the accumulator. (1 per -)	**AO1.2** **2**	
	b		Direct addressing	**AO1.2**	Accept Symbolic Addressing

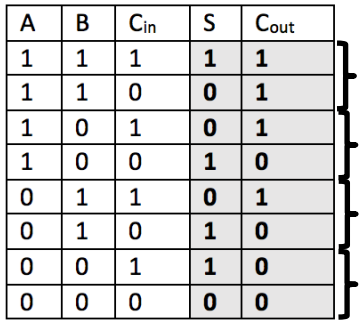
				1	
	c		- Answer contains at least 1 followed by 0 - Answer contains at least three 10s - Answer contains exactly four 10s (1 per -)	**AO3.3** **3**	1 0 1 0 1 0 1 0 NB allow answers that are vertical or horizontal.
	d		- The address of one is stored in the MAR - This value is sent along the address bus AND the fetch signal is sent on the control bus. - The contents of one are sent from memory to the processer on the data bus and stored in the MDR - The contents of the MDR and ACC are sent to the ALU - The result is stored back in the ACC (1 per -)	**AO2.2** **5**	Accept MBR instead of MDR
	e		- An instruction can be fetched as the previous one is being decoded… - …and the one before that is being executed. - E.g. `LDA Zero` can be fetched, while `OUT` is being	**AO1.2** **(2)**	

			decoded and `start LDA one` is being executed. (1 per -)	**AO2.2** **(1)** **3**	
	f		- BRP could be followed by one of two possible instructions, which one will only be determined at execution - Meaning the wrong one may be fetched/decoded (1 per -)	**AO2.2** **2**	
	g		- Clock speed - Cache Size - Number of cores (1 per max 1)	**AO1.1** **1**	

6	a		- Adds the tag name…
			- Includes the opening <
			- Includes the closing > and nothing further
			- Tags are added to data structure.
			- Adds all tags in the string.
			- Sensible variable names used
			- Correct use of indentation
			(1 per -)
				AO3.2
				
				7
					`tagStartPos = 0`
					`insideTag = false`
					`i = 0`
					`while i < htmlCode.length`
					`  if htmlCode.substring(i,1) == ”<” and insideTag == false then`
					`  tagStartPos = i`
					`  insideTag = true`
					`  elseif insideTag == true and htmlCode.substring(i,1) == ”>” then`
					`  dataStructureA.add(htmlCode.substring(ta`

					gStartPos, i-tagStartPos+1))
					insideTag = false
					endif
					i = i + 1
					endwhile

	b	i	Queue	**AO2.1** **1**	
		ii	Hashtable	**AO2.2** **1**	Accept Hashmap/Associative Array/Dictionary
		iii	Stack	**AO2.2** **1**	
		iv	Stack uses a last in first out approach… … and the last HTML tag to be opened should be the first to be closed.	**AO2.2** **2**	
7	a			**AO1.2** **2**	

			1 mark for the first two rows 1 mark for the last two rows		
	b	i	 1 Mark for rows 1 and 2 1 Mark for rows 3 and 4 1 Mark for rows 5 and 6 1 Mark for rows 7 and 8	**AO1.2** **4**	
		ii	- Circuit adds two bits (and a carry bit) together/is an adder. - A B and C_{in} are added together - The result is given in S - And a carry bit in C_{out} (1 per -)	**AO2.2** **4**	

	c	i	$S \equiv A \underline{\vee} B \underline{\vee} C_{in}$	**AO2.2** **1**	Accept XOR instead of $\underline{\vee}$ Accept $\oplus$ instead of $\underline{\vee}$
		ii	$C_{out} \equiv ((A \underline{\vee} B) \wedge C_{in}) \vee (A \wedge B)$ One mark for $((A \underline{\vee} B) \wedge C_{in})$ One mark for $\vee (A \wedge B)$	**AO2.2** **2**	Accept XOR instead of $\underline{\vee}$ Accept $\oplus$ instead of $\underline{\vee}$ Accept AND instead of $\wedge$ Accept OR instead of $\vee$ Accept + instead of $\vee$
8	a		- A primary key must have a unique value for every record - The values for all these fields could repeat. (1 per -)	**AO1.1** **(1)** **AO1.2** **(1)** **2**	
	b		- Advantage: Searches of Artist can be performed more quickly. - Disadvantage: The index takes up extra space in the database. (1 per -)	**AO1.2** **2**	

	c		- Song and `Playlist` would have a many to many relationship - This is not allowed - Adding a table between them resolves this - Diagram to illustrate this.  (1 per -)	**AO3.1** **4**	
	d	i	Copyright, Design and Patents Act	**AO1.1** **1**	Accept Copyright Act/Law
		ii	- `DELETE FROM Song` - `WHERE Artist='RandomBits'` (1 mark per -, max 2)	**AO3.1** **2**	
		iii	All entries in `PlayListEntry` which contain songs by RandomBits must be removed.	**AO2.1** **1**	
9			**Mark Band 3–High Level (9-12 marks)** The candidate demonstrates a thorough knowledge and understanding of the technical and legal aspects of privacy. The material is generally accurate and detailed.	**AO1.1** **(2)** **AO1.2** **(2)**	**AO1 Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: - Modern encryption is easy to access. - The strongest encryption is (as far as is known). unbreakable

			The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up both sides of the argument which results in a supported and realistic judgment as to how achievable privacy is. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level (5-8 marks)** The candidate demonstrates reasonable knowledge and understanding of technical and legal aspects of privacy; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation.	**AO2.1** **(3)** **AO3.3** **(5)** **12**	even by government agencies. - People leave a digital footprint wherever they go (mobile phones can track our location, store cards record our shopping habits). - Our online activities can be tracked by IP address and 3rd party cookies. - CCTV is ubiquitous. Most people carry round phones capable of taking video/photos. - Facial recognition AI techniques mean we may be filmed whilst unaware and subsequently identified. - The Data Protection Act aims to protect people's data. - Computer Misuse Act Prosecutes those gaining unauthorised access to computer systems which may deter attempts to gain unauthorised access to data. - The Regulation of Investigatory Powers Act regulates how the authorities can monitor our actions. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. Examples may include but are not limited to: People can secure their data using encryption but the Regulation of Investigatory Powers Act can force them to share their key with the authorities. The government is becoming increasingly worried about encryption and there is the possibility of laws to limit its use in the future.

			The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine how achievable privacy is. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-4 marks)** The candidate demonstrates a basic knowledge of the technical and legal aspects of privacy with limited understanding shown; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence*		Laws such as the Computer Misuse Act can act as a deterrent but the Internet is global and it is hard to pursue and prosecute offenders from other countries. Trying to be in a location anonymously is very difficult as movements are tracked in many different ways and this is becoming increasingly automated. **AO3.3: Evaluation** Having considered the different sides to the argument candidates will need to reach a supported judgment based on the evidence included in their response. There should be no bias in marks as to which viewpoint is chosen but especially in the top mark band there must be a clear link between the points candidates have made and justification.

			may not be clear. **0 marks** No attempt to answer the question or response is not worthy of credit.		
10	a		To control the hardware	**AO1.1** **1**	
	b		- Developers can run their operating system on a software implementation of the phone… - …Until the physical machine is ready. (1 per -)	**AO1.2** **2**	
	c		- Complete the current FDE Cycle - Check the priority of the incoming interrupt. - If its of a higher priority than the current task. - Contents of registers stored in memory.. - …in a stack. - The relevant interrupt service routine is loaded… - ..by loading the relevant value into the program counter. - When the ISR is complete the previous state is popped from the stack - And are loaded back into the registers. (1 per -, max 6)	**AO1.1** **6**	

	d	i	- Feasibility Study - Investigation/Requirements Elicitation - Analysis - Design - Implementation/Coding - Testing - Installation - Documentation - Evaluation - Maintenance (1 per -, max 3)	**AO1.1** **3**	
	d	ii	- Tends to suit large scale projects… - ..An OS is an example of such a big project. - Tends to suit projects with stable requirements… - …And the base requirements of an OS are unlikely to change. (1 per -, max 2)	**AO1.2** **2**	
		iii	If a change does occur in the requirements the lifecycle cannot respond easily, often at the cost of time and money.	**AO1.1** **1**	
	e		**Mark Band 3–High Level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of Object Oriented Programming and has discussed inheritance, polymorphism and encapsulation; the material is generally accurate and detailed.	**AO1.1** **(2)** **AO1.2** **(2)** **AO2.1**	**AO1 Knowledge and Understanding** The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive: OOP involve solutions being constructed by means of objects that interact with each other. OOP uses classes as templates to

			The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level (4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of a range of Object Oriented Programming and has discussed at least two of: inheritance, polymorphism and encapsulation; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation.	**(2)** **AO3.3** **(3)** **9**	construct objects. An object has attributes (variables associated with it) and methods (subroutines that form the actions an object can carry out). Inheritance is where a class retains the methods and attributes of its parent class as well as having its own. Encapsulation is the process of keeping an object's attributes private so they can only be accessed and changed via public methods. Polymorphism means that objects of different types can be treated in the same way. Procedural programming breaks a solution down into subroutines. These subroutines are re built and combined to form a program. **AO2.1: Application** The selected knowledge/examples should be directly related to the specific question. Examples may include but are not limited to: Breaking a problem down into objects naturally lends itself to teams as different team members can work on different objects. Inheritance means that one class can be coded and that code used as the base for similar objects. This will save the team time as they are able to build on work already done. Encapsulation means that objects only interact in the way intended and prevents unexpected changed to attributes having unforeseen consequences. This means there are likely to be fewer issues as the team combines their code.

			The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-3 marks)** The candidate demonstrates a basic knowledge of Object Oriented Programming with limited understanding shown; the material is basic and contains some inaccuracies. For 3 marks they have described at least one of inheritance, polymorphism or encapsulation. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be*		Polymorphism means that code can be written that is able to handle different objects in the same way. This reduces the volume of code the team need to produce. Procedural programming can be divided between a team with different team members tackling different subroutines. There are a number of similarities between the two paradigms. Certain problems lend themselves more to one than the other. **AO3.3: Evaluation** Having considered the different sides to the argument candidates will need to reach a supported judgment based on the evidence included in their response.

			clear. 0 marks No attempt to answer the question or response is not worthy of credit.		
11	a		`<img src=”UKstamps.jpg”>` - One mark for img tag - One mark for correct src attribute **(1 per -)**	**AO3.1** **2**	Accept self closing tag:
	b		- `<a> </a>` tags plus `Find out more about UK stamps` text between them. - `href` attribute with value `http://ukstampcollectorsguild.co.uk` (1 per -)	**AO3.1** **2**	`<a href=”http://ukstampcollectorsguild.co.uk”>Find out More about UK stamps</a>`
	ci		- Formatting code has to be rewritten for every page - Changes have to be made to every page - It is a lot of work to keep the look of the site consistent. `(1 per -, max 1)`	**AO1.2** **1**	

	ii		The site is slower to access (as the formatting information is reloaded for every page) Unlikely to have formatting specific to their device/needs.	**AO1.2** **1**	
	d		- Change the tag body… - So it includes the attribute style - Which should have the value `background-color:lightGreen` (1 per -)	**AO2.2** **3**	Accept: `<body bgcolor="lightGreen">` for full marks
	e		- The number of sites that link to their site - The PageRank of the linking sites - The number of outward links from the site (1 per -, max 2)	**AO1.2** **2**	

OCR
Oxford Cambridge and RSA

A Level Computer Science

H446/01 Computer Systems

Practice paper – Set 2

Time allowed: 2 hours 30 minutes

Do not use:
- a calculator

First name									
Last name									
Centre number						Candidate number				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided. Additional paper may be used if required but you must clearly show your candidate number, centre number and question number(s).
- Do **not** write in the barcodes.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **32** pages.

Answer **all** questions.

1 An operating system has to manage a system's resources.

(a) One aspect of this is memory management.

(i) Describe **one** difference between paging and segmentation.

.....

.....

.....

..... **[2]**

(ii) Explain how an operating system may overcome the problem of physical memory being full.

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(b) Another job of an operating system is to deal with interrupts.

(i) State what is meant by the term 'interrupt'.

.....

..... **[1]**

(ii) Describe what happens in the CPU when it receives an interrupt.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

2 Mobile Treasure Hunt is a game played on a mobile phone. The game shows the user's position on a map of their local area. Treasure randomly appears on the map and users must move to the appropriate area to collect the treasure before it disappears.

(a) State the name of a sensor or input device the phone might use when playing Mobile Treasure Hunt and explain why it might be used.

Sensor / Input Device: .....

Use: .....

..... **[2]**

Below is part of the code from Mobile Treasure Hunt.

```
class Treasure

    private value
    private weight
    private name

    public procedure new(givenName)
        name=givenName
        weight=20
        value=randomInteger(1,20)
    endprocedure

    public procedure changeName(givenName)
        name=givenName
    endprocedure

endclass

class TreasureChest inherits Treasure

    private locked

    public procedure new(givenName)
        super.new(givenName)
        locked=false
        value=randomInteger(1,100)
        weight=randomInteger(80,120)
    endprocedure

    public procedure pickLock()
        if getRandomNumber()>0.5 then
            locked=false
        endif
    endprocedure

endclass
```

Fig. 2.1

(b) Explain what is meant by the term 'encapsulation' with reference to the attribute called `name`.

.....

.....

.....

.....

.....

..... **[3]**

(c) Describe what is meant by the term 'inheritance', referring to the code in Fig. 2.1.

.....

.....

.....

.....

.....

..... **[3]**

(d) Identify all attributes and methods in the `TreasureChest` class.

Methods: .....

.....

Attributes: .....

..... **[2]**

3 A Little Man Computer (LMC) assembly language program is stored in memory as shown in Fig. 3.1.

0	LDA &7
1	ADD #4
2	OUT
3	HLT
4	6
5	2
6	10
7	15
8	16
9	17

Fig. 3.1

In this variant of LMC the symbols & and # are used to denote different modes of addressing.

(a) Given that the output is 17, state the addressing mode represented by each symbol.

(i) & ..... **[1]**

(ii) # ..... **[1]**

An assembler is used on the code.

(b) Describe what is meant by the term 'assembler'.

.....

.....

.....

..... **[2]**

(c) Explain how pipelining would help a CPU execute the code in Fig. 3.1 more quickly.

.....

.....

.....

.....

.....

..... **[3]**

4 A bus runs between two cities. There are a number of stops on the bus route labelled `StopA`, `StopB` and so on. The timetable for the route is represented as a hash table. For each entry in the hash table the key is the bus stop code and the data attached to it is a (zero indexed) array of the times a bus arrives at the stop. The times are stored as strings.

An extract of the hash table is shown below:

```
times=
{
"StopA":["06:55", "07:25", "07:55", "08:55", "09:55", "11:55", "14:00",
"15:00", "15:30", "16:00"]
"StopB":["06:40", "07:40", "08:40", "09:20", "09:40", "14:00", "15:00",
"16:00", "16:30"]
…
…
}
```

`print(times["StopA"][1])` displays `07:25`

(a) State what the code `print(times["StopB"][4])` displays.

..... **[1]**

(b) Write a function called `timeValue` that given a time stored in a string, returns the equivalent integer (using thousands and hundreds for the hours and tens and units for the minutes). The given string should be assumed to represent the time in the 24-hour clock in the format HH:MM

`timeValue("07:55")` should return `755`
`timeValue("15:30")` should return `1530`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

(c) Write code for a function that takes in the name of a stop (`stopName`) and the current time as an integer (`currentTime`) in the format described in part (b) (using thousands and hundreds for the hours and tens and units for the minutes). It should return the time of the next available bus in the string format. If there are no more buses available that day it should return the string "No buses".

Example `nextBus("StopA", 1013)` should return `"11:55"`

```
function nextBus(stopName, currentTime)
```

.....

.....

.....

.....

.....

```
endfunction
```

[5]

5 Every bank account has an account number and sort code. The sort code identifies the bank branch (location of the bank) with which the account is held and the account number uniquely identifies the bank account. An extract from a bank's database table is shown in Fig. 5.1.

CustomerID	Forename	Surname	Acc No	Sort Code	Branch Name
145204	Elaine	Murray	14725200	67-34-56	Hull
657875	Jordan	Rogers	62703441	67-45-67	Truro
735951	Monim	Khan	96385547	67-00-11	Cambridge
744078	Tom	Banner	45623929	67-00-11	Cambridge

Fig. 5.1

(a) State why the table in Fig. 5.1 is not in Third Normal Form.

.....

..... **[1]**

(b) Explain how the database could be put into Third Normal Form.

.....

.....

.....

.....

.....

..... **[3]**

(c)* The bank needs to ensure the data stored in its database is accurate at all times including when customers deposit or withdraw funds.

Discuss how the bank can ensure the accuracy of its data and the importance of doing so.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

6 The XOR operator can be used to encrypt data.

(a) Show the effect of applying XOR on Text and Key, by completing the last row of the table below.

Text	O								C								R							
Value	0	1	0	0	1	1	1	1	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0
Key	A								B								C							
Value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1
XOR																								

[2]

(b) Show the effect of applying XOR on your answer to part (a) and Key, by completing the first and last rows of the table below.

(a)																								
Key	A								B								C							
Value	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1
XOR																								

[2]

(c) Explain whether the type of encryption described above is symmetric or asymmetric.

.....

.....

.....

..... **[2]**

(d) Explain why asymmetric encryption is more suited to transactions over the internet than symmetric encryption.

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

7 A binary search tree is used to store the names of dog breeds.

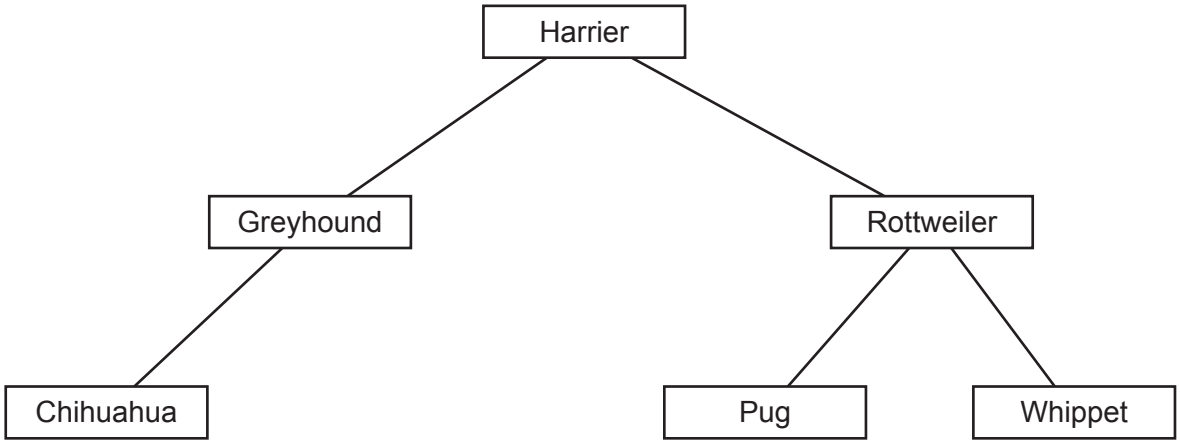


Fig. 7.1

(a) The breeds Doberman and Dalmatian are added to the tree in that order. Add them to Fig. 7.1. **[2]**

(b) Explain how you would determine if the breed Pug is in the binary search tree.

.....

.....

.....

.....

.....

..... **[3]**

(c) Explain how you would determine if the breed Spaniel is in the binary search tree.

.....

.....

.....

.....

.....

..... **[3]**

(d) The tree is coded using object oriented programming.

Each dog breed is represented by an object of class Node.

The Node class has the methods:
`getLeftNode()` – returns the left hand child node or null if there is no left hand child.
`getRightNode()` – returns the right hand child node or null if there is no right hand child.
`getBreed()` – returns the name of the breed stored in that node.

The program allows for a breed name to be entered, and depending on whether the breed is in the tree or not, displays either:

<breed name> is not in the tree.

or

<breed name> is in the tree.

Complete the program below. Credit will be given for readability of code.

```
name=input("Enter the name of a breed")
breedNode=tree.root() //breedNode is an object of type Node
                      //representing the root of the tree
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

8 A website has the following code.

```
<p>
<form action="checkUser.php">
    Username:<br>
    <input type="text" name="username">
    <br>
    Password:<br>
    <input type="password" name="password">
    <br><br>
    <input type="submit" value="Submit">
</form>

<p id="warning">Unauthorised access to this system will be
prosecuted</p>
```

The page is linked to a style sheet. The message `Unauthorised access to this system will be prosecuted` is red with a monospace font. (Note this is the only text on the page that has this formatting)

(a) Write the segment of CSS code that would appear on the style sheet to make the message appear in the way described.

.....

.....

.....

.....

.....

..... **[3]**

(b) Explain the meaning of the HTML line `<input type="text" name="username">`

.....

.....

.....

..... **[2]**

(c)* The line `<form  action="checkUser.php">` sends the contents of the form to be processed by the server. This is done by code written in a language called PHP which is designed for server side processing. Conversely JavaScript is traditionally used for client side processing.

Discuss the difference between server and client side processing with respect to webpages. You should refer to the advantages, drawbacks and best uses of both approaches.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[9]**

Part of the code on the server can be represented in pseudocode below.
In the pseudocode:
`RunSQL(A,B)` runs SQL statement `A` on database `B`. In this case it will always return a single value.
`valueFromForm(controlName)` gets the value entered into the input control with the name `controlName`

```
01 username = valueFromForm("username")
02 password = valueFromForm("password")
03 statement = "SELECT passwordHash FROM users WHERE name = '"
   + username + "'"
04 hashInDB = RunSQL(statement, siteDatabase)
05 if hashInDB == hash(password) then
06     //TO ADD: Generate success webpage
07 else
08     //TO ADD: Store IP address of user in database.
09 endif
```

Fig. 8.1

(d) Explain what the code in Fig. 8.1 does.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(e) In certain scenarios the user's IP address is logged in a database.

(i) Describe what is meant by an IP Address.

.....

.....

.....

..... **[2]**

(ii) Explain why the programmers have chosen to store the user's IP address.

.....

.....

.....

..... **[2]**

(f) An extract from the database is shown below:

userID	**name**	**passwordHash**
1	admin	0e5a511
2	DenverJ34	f60ccdc
3	TaylorJ22	3a050bc

(i) The username `admin` is entered into the form.

State what the value of `statement` would be after line 03 of the code in Fig. 8 .1 is run.

.....

..... **[1]**

(ii) State what the value of `hashInDB` would be after line 04 of the code in Fig. 8.1 is run.

.....

..... **[1]**

(g) In SQL the character `;` denotes the next statement and the characters `--` denote a comment.

The username `DenverJ34'; DROP TABLE users; --` is entered into the form.

(i) State what the value of `statement` would be after line 03 is run.

.....

.....

.....

..... **[1]**

(ii) Describe what happens when line 04 is run.

.....

.....

.....

..... **[2]**

(iii) State the name of a law the user has broken by entering the username
`DenverJ34'; DROP TABLE users; --`

.....

..... **[1]**

9

CD \ AB	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	1	0
10	1	1	1	0

Fig. 9.1

(a) State the Boolean expression represented by the Karnaugh map in Fig. 9.1, in its smallest form.

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(b) State the simplified versions of the following Boolean expressions:

(i) ¬ ¬A

.....

..... **[1]**

(ii) (¬A ∧ ¬B)

.....

..... **[1]**

(iii) ¬ (¬A ∧ ¬B)

.....

..... **[1]**

10 A NAND gate and its truth table are shown in Fig. 10.1.

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

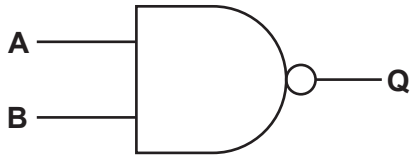


Fig. 10.1

(a) Draw a set of gates equivalent to a NAND gate, but built only of AND, OR and NOT gates.

[2]

The component below is a D-Type, positive edge triggered, flip-flop.

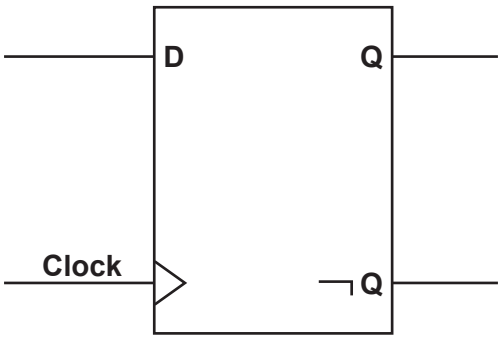


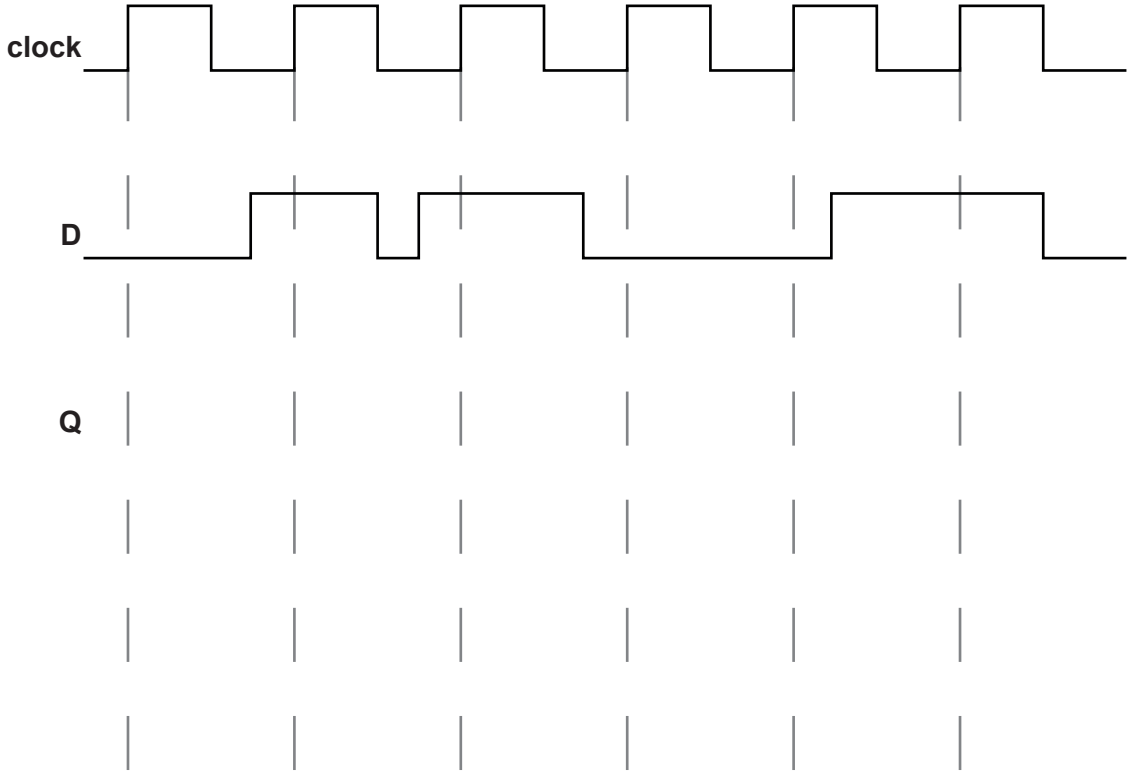
Fig. 10.2

(b) State the purpose of a flip-flop.

.....

..... **[1]**

(c) Draw the output of the flip-flop from Fig. 10.2 on the diagram below.



[3]

11 **(a)** Show a representation of the hexadecimal number AB in:

(i) Binary

.....

.....

.....

..... **[1]**

(ii) Denary

.....

.....

.....

..... **[1]**

(b) Show a representation of denary -119 in 8-bits using:

(i) Sign and Magnitude

.....

.....

.....

..... **[1]**

(ii) Two's Complement

.....

.....

.....

..... **[1]**

(c) A floating point number is represented with a mantissa of 8-bits followed by an exponent of 4-bits, both in two's complement.

00011010 0010

(i) Identify whether or not the number is normalised.

..... **[1]**

(ii) State how you arrived at your answer to part (i).

.....

..... **[1]**

(d) Two floating point numbers are shown below in the same format as used for part (a). Calculate the answer of the second number subtracted from the first. You must show your working and ensure your answer is normalised.

01001100 0011 - 01001010 0010

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

12* Some problems require a large amount of computing power that goes well beyond a single CPU.

Discuss the different approaches that can be taken to provide increasingly larger amounts of computing power and the types of problem they are suited to.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[12]**

END OF QUESTION PAPER

BLANK PAGE

BLANK PAGE

BLANK PAGE

OCR
Oxford Cambridge and RSA

Practice Paper 2

GCE Computer Science

H446/01 Computer Systems

Duration: 2 hours 30 minutes

MAXIMUM MARK 140

Final

This document consists of 30 pages

Question			Answer	Marks	Guidance
1	a	i	Paging is fixed size / physical divisions (1) Segmentation is dividing memory logically (i.e. will be variable size). (1)	**2 (AO1.1)**	
		ii	Uses virtual memory (1) which is an allocated area of a hard drive/secondary storage device.(1) Pages that aren't needed are moved to VM (1) and moved back to physical memory when required (1)	**4 (AO1.2)**	
	b	i	A signal to the processor indicating a device/process needs attention (1)	**1 (AO1.1)**	
		ii	Any five from: If the interrupt is of a lower/equal priority to the current process then the current process continues (1) If it is of a higher priority the CPU finishes its current Fetch-Decode-Execute cycle (1). The contents of the CPU's registers are copied to a stack (1) in memory (1). The location of the appropriate interrupt service routine is loaded into the program counter (1) When the ISR is complete, the previous contents are popped from the stack and loaded back into the registers (1).	**5 (AO1.2)**	
2	a		Any two from: GPS…(1) …To determine the user's geographical location (1) Compass/magnetometer (1)… …To determine direction in which use is facing. Accelerometer…(1) …to recognise user's movement. (1) Touchscreen…. (1) …. To select options/play the	**2 (AO1.2)**	

			game (1)		
	b		When an attribute is made private (so it can't be directly accessed or changed from outside the class) (1) Public methods are used to read / amend the attribute's value (1) The attribute name's value can only be amended through the method `changeName`. (1)	**3 (AO 1.2)**	
	c		When a class has the attributes and methods of its parent class. (1) It may also have methods and attributes of its own (1) `TreasureChest` inherits from the class `Treasure` (1)	**3 (AO 1.2)**	
	d		Methods: (constructor/`new`), `changeName`, `pickLock` (1) Attributes: `value`, `weight`, `name`, `locked` (1)	**2 (AO 1.2)**	Do not penalise for not including constructor. Only give method mark if both other methods are listed Only give attributes mark if all four attributes are listed.
3	a	i	& immediate addressing	**1 (AO2.2)**	
		ii	# indirect addressing	**1 (AO2.2)**	
	b		A program that translates assembly code (1) into machine code/object code (1)	**2 (AO1.1)**	
	c		Pipelining would allow one instruction to be fetched as the previous one is being decoded and the one before that is being executed.(1) For example `OUT` could be fetched (1) . As there are no jump/branch instructions it pipelines well (as there is no need to flush the pipeline). (1)	**3 (2 AO2.2, 1 AO3.2)**	Accept any valid example from the given code.
4	a		09:40	**1**	

<table>
<tr><td></td><td></td><td></td><td></td><td>(AO1.2)</td><td></td></tr>
<tr><td></td><td>b</td><td></td><td>Correctly named function that takes in time as a parameter and returns a value. (1)
Minutes element is correct (1)
Hours element is correct (1)</td><td>3

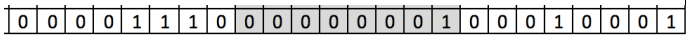
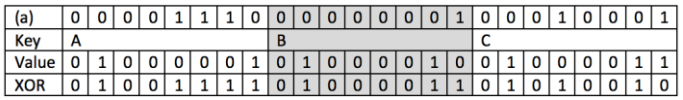
(AO 3.2)</td><td>Returned value needn’t be correct for first mark
Example solution:
<pre>function timeValue(givenTime)
    intTime=int(givenTime.substring(3,2))
    intTime=intTime+int(givenTime.substring(0,2))*100
    return intTime
endfunction</pre></td></tr>
<tr><td></td><td>c</td><td></td><td>Correct stop array extracted/referenced in code (1)
Sensible attempt to iterate through the array (1)
Program returns time of next bus (1)
Program returns No Buses when no more buses left. (1)
Program runs without an index out of bounds error (You may assume short circuit evaluation i.e. if the array is in the second part of an and condition it won’t be checked if the first half evaluates to false.) (1)</td><td>5

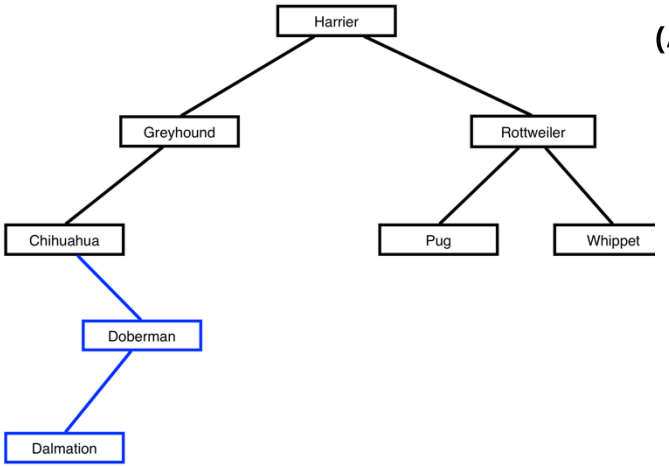
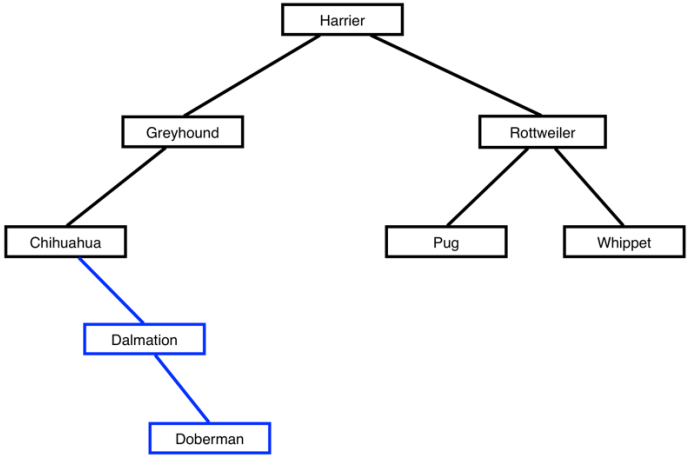
(A03.2)</td><td>Marks 1-2 can be awarded even if the program doesn’t exhibit behaviour needed for marks 3-5
Marks 1-4 can be awarded even if mark 5 can’t be awarded as program would in reality crash.

<pre>count = 0
timesLeft = true
timesList = times[stopName]
while timesLeft == true and
timeValue(timesList[count]) < currentTime
   count = count + 1
   if count == timesList.length then
       timesLeft = false
   endif
endwhile
if timesLeft == true then
   return timesList[count]
else
   return "No Buses"
endif</pre></td></tr>
<tr><td>5</td><td>a</td><td></td><td>Branch name depends on Sort Code (i.e. there is a transitive relationship).</td><td>1
(AO2.1)</td><td></td></tr>
<tr><td></td><td>b</td><td></td><td>Create another table for Branches which should</td><td>3</td><td></td></tr>
</table>

			include sort-code and branch name. (1) Make sort code the primary key of the BRANCH table/ Add a primary key to BRANCH. (1) Remove Branch name from Customers, leave sortcode as primary key/ Remove sort-code and branch name from customers and add the primary key values from BRANCS as the foreign key (1) ALTERNATIVE ANSWER (ER-DIAGRAM) Two tables CUSTOMER and BRANCH (or similar names) (1) Link from CUSTOMER to BRANCHES is Many (1) to One (1)	**(AO3.1)**	
	c		**Mark Band 3–High Level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of transaction processing. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and*	**2 (AO1.1)** **2 (AO1.2)** **2 (AO2.1)** **3 (AO3.3)**	*Answers may include, but are not limited to, some of the points below.* **AO1: Knowledge and Understanding** Transactions should be: Atomic; They should either succeed or fail but never partially succeed. Consistent: The transaction should only change the database according to the rules of the database. Isolated: Each transaction shouldn't affect/overwrite other transactions concurrently being processed. Durable: Once a transaction has been started it is remains no matter what happens. Records should be locked when in use. If one transaction is amending a record, no other transaction should be able to until the first transaction is complete. Transactions should maintain referential integrity. Changes to data in one table must take into account data in linked tables. Data should have redundancy – if part of a database is lost it should be recoverable from elsewhere.

			substantiated. **Mark Band 2-Mid Level (4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of transaction processing; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-3 marks)** The candidate demonstrates a basic knowledge of transaction processing; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made		Data entered must be accurate in the first place. Security measures need to be in place to prevent malicious tampering of data. Data entered should be validated (automatically checked it is sensible) and verified (checked that the data entered matches the original). **AO2.1: Application** Ensuring the accuracy of transactions will be partly down to the DBMS and partly down to the code accessing the DBMS. Referential Integrity is often enforced by the database management system. Redundancy can be provided in a number of ways. This could be a RAID setup or mirroring servers. Bank may use validation and verification when data is input. Security procedures may include firewall, enforcement of sensible passwords and enforced user access rights. Validation may include range checks, list checks, presence checks etc. Verification may include double entry and proof reading, **AO3.3: Evaluation** It is essential the bank follows the precautions discussed. Verification and validation help ensure the data is initial data is sound (garbage in = garbage out) If they make mistakes with their financial data they may lose money or overcharge customers and lose business/find themselves in legal trouble. Without redundancy data could be lost. Without careful transaction processing, one transaction could accidentally overwrite another or half complete leading to inaccurate data. Under the Data Protection Act they have an obligation to keep personal data accurate.

			are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		Verification and Validation.
6	a		 One byte correct (1) all three bytes correct.(1)	**2 (AO 1.2)**	
	b		 One byte correct (1) all three bytes correct.(1)	**2 (AO 1.2)**	Allow FT if (a) is incorrect but bottom row must match XOR with top row and key.
	c		Symmetric (1) …. as the same key is used to decrypt it as encrypt it (1)	**2 (1 - AO1.2, 1 - AO 2.2)**	Allow FT for asymmetric if (b) indicates asymmetric encryption used
	d		Any four from: Symmetric encryption would require both parties to have copy of the key (1) this couldn't be transmitted over the internet or an eavesdropper monitoring the message may see it (1) Asymmetric gets round this requirement as there are two different keys (1) One key encrypts the data (1) which can be publically distributed (1) and a different key to decrypt it (1) which is kept private (1)	**4 (AO 1.2)**	

7	a		 Doberman in correct position (1) Dalmatian in correct position (1) (Allow FT if first mark is incorrect)	**2** **(AO 2.2)**	Allow one mark if added in wrong order. 
	b		Pug > Harrier (go right) (1) Pug < Rottweiler (go left) (1) Found Pug (1)	**3** **(AO 2.2)**	
	c		Spaniel > Harriet (go right) (1) Spaniel> Rottweiler (go right) (1)Spaniel < Whippet, no child node so Spaniel is not in tree (1)	**3** **(AO 2.2)**	

<table>
<tr>
<td></td>
<td>d</td>
<td></td>
<td>Calls <code>getLeftNode()</code> when name is less than the value of the current node (1)
and calls <code>getRightNode()</code> when name is less than the value of the current node. (1)
Declares a breed to be in the tree if and only if it exists.(1)
Declares a breed not to be in the tree if and only if it doesn't exist (1)
Presents output strings in correct format (1)
Sensible use of variable names and correctly indented (1)</td>
<td>6
(5 AO 3.2, 1 AO 1.2)</td>
<td>Points 4 and 5 can be awarded even if 1-3 aren't.

<pre>notThere = false

while breedNode.getName() != name and notThere == false
      if name < breedNode.getName() then
            if breedNode.getLeftNode() != null then
                  breedNode = getLeftNode()
            else
                  notThere = true
            endif
      else // must be greater
            if breedNode.getRightNode() != null then
                  breedNode = getRightNode()
            else
                  notThere = true
            endif
      endif
endwhile

if notThere == true then
      print(name+ " is not in the tree.")
else
      print(name+" is in the tree")
endif</pre></td>
</tr>
<tr>
<td>8</td>
<td>a</td>
<td></td>
<td>Code enclosed within <code>#warning{…}</code> (1)
<code>color: red;</code> (1)
<code>font-family: monospace;</code> (1)</td>
<td>3
(AO 3.1)</td>
<td><pre>#warning{
color: red;
font-family: monospace;
}</pre>Also accept hex color and RGB color notations.
Don't penalise for missing semicolons.

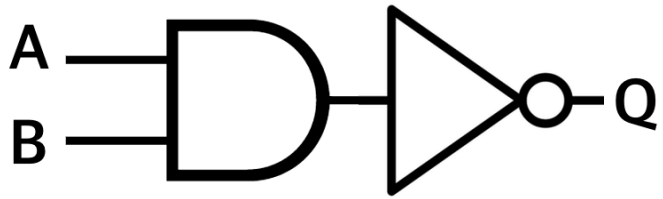
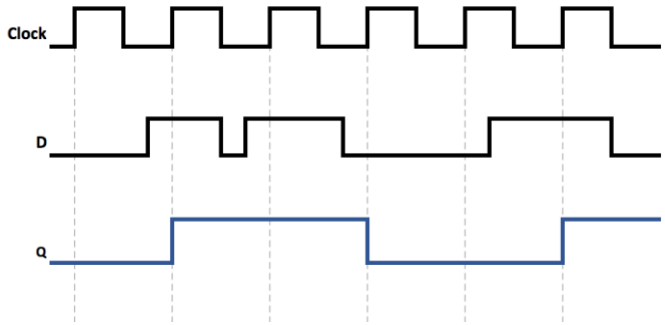
Accept a named suitable font like <code>Courier New</code>.</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td></td>
<td>Creates a textbox (1)
To hold the username/which is referred to as</td>
<td>2
(AO 2.2)</td>
<td></td>
</tr>
</table>

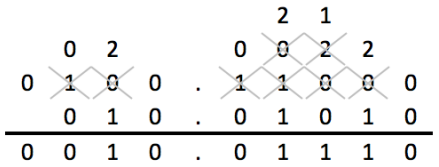
			username (1)		
	c		**Mark Band 3–High Level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of client and server side processing. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well-considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level (4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of client and server side processing; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a sound discussion, the majority of which is focused. Evaluative comments are for the most part appropriate, although one or	**2 (AO1.1)** **2 (AO1.2)** **2 (AO2.1)** **3 (AO3.3)**	*Answers may include, but are not limited to, some of the points below.* **AO1: Knowledge and Understanding** Server side processing takes place on the webserver. Data is sent from the browser to the server, the server processes it and sends the output back to the browser. Client side processing takes place in the web browser. **AO2.1: Application** Client side processing doesn't require data to be sent back and forth meaning code is much more responsive. Code is visible which means it can be copied. The browser may not run the code either because it doesn't have the capability or because the user has intentionally disabled client side code. Server side processing takes away the reliance of the browser having the correct interpreter. It hides the code from the user, protecting copyright and avoiding it being amended/circumvented. Server side processing puts extra load on the server. This is at the cost of the company hosting the website. **AO3.3: Evaluation** Client side processing is best used when it's not critical code that runs. If it is critical then it should be carried out on the server. Client side processing is also best where a quick response is needed – an example being games. Server side processing is best used where it is integral that processing is carried out. It is often used for generating content. It can be used to access data including secure data. For this reason any data passed to it has to be checked carefully. With some things like validation good practice is to do both: First on the client for a quick response if there is an issue, then on the server

			two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-3 marks)** The candidate demonstrates a basic knowledge of client and server side processing; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgments if made are weak and unsubstantiated. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		in case the client side processing has been circumvented.

	d		Any five from: Takes the `username` and `password` from the form (1) Uses the username to create an SQL statement (1) to get the `passwordHash` belonging to the given `username` (1) Runs the SQL Statement(1) hashes the given `password` and compares it to the retrieved hash (1) If they match it generates a success webpage, otherwise it records the user's IP address. (1)	**5 (3 AO2.2, 2 AO3.3)**	
	e	i	Any two from: A numerical address made of 4 numbers each between 0 and 255 / 32 hexadecimal digits (1) That uniquely identifies a device on a network. (1) It is a logical identifier (i.e. can change on a physical device) (1)	**2 (AO2.1)**	

		ii	IP address can help identify a user… (1) …so company can potentially track users attempting to gain unauthorised access (1)	**2 (AO 2.2)**	
	f	i	`SELECT passwordHash FROM users WHERE name = 'admin'`	**1 (AO 1.2)**	
		ii	`0e5a511`	**1 (AO 1.2)**	
	g	i	`SELECT passwordHash FROM users WHERE name = 'DenverJ34'; DROP TABLE users; --'`	**1 (AO 1.2)**	
		ii	Gets `passwordHash` for username DenverJ34 (1) then deletes the table called users. (1)	**2 (AO 3.3)**	
		iii	Computer Misuse Act	**1 (AO 1.1)**	
9	a		(¬A ∧¬D) ∨(A ∧B ∧C) ∨(¬B ∧¬C ∧¬D) One mark for each bracketed section. One mark for them being joined with ORs	**4 (AO 1.2)**	
	b	i	A	**1 (AO 1.1)**	
		ii	¬(A∨ B)	**1 (AO 1.1)**	
		iii	A∨B	**1 (AO 1.1)**	

10	a		 One AND one NOT gate used (1) In correct configuration (1)	**2 (AO 1.2)**	
	b		To store the state of a bit	**1 (AO1.1)**	
	c		 One mark for each two correct clock cycles.	**3 (AO2.2.)**	
11	a	i	10101011	**1 (AO1.2)**	
		ii	171	**1 (AO 1.2)**	
	b	i	11110111	**1 (AO 1.2)**	

		ii	10001001	**1 (AO 1.2)**	
	c	i	Not Normalised	**1 (AO 1.2)**	
		ii	(Mantissa) Starts with 00 (normalised numbers start 01 or 10)	**1 (AO 1.2)**	
	d		Exponent of first number is 3 (1) Making it 0100.1100 (1) Exponent of second number is 2 (1) Making it 010.01010 (1)  Subtract numbers (1) Normalised is 01001110 0010 (1)	**5 (AO 1.2)**	Accept any sensible method (eg converting one number to have same exponent as other and subtracting) with correct answer for full marks.
12			**Mark Band 3–High Level (9-12 marks)** The candidate demonstrates a thorough knowledge and understanding of methods of utilising large amounts of computing power. The material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up both sides of the argument which results in a supported and realistic judgment as to which approaches to provide increasingly larger amounts of computing power are	**2 (AO1.1)** **2 (AO1.2)** **3 (AO2.1)** **5 (AO3.3)**	*Answers may include, but are not limited to, some of the points below.* **AO1: Knowledge and Understanding** Processors have increasingly large clock speeds and can be overclocked. Processors can have multiple cores. Super computers can have multiple processors (and GPUs). GPUs can be applied to problems other than graphics processing. Problems can be distributed across a number of computers working together. **AO2.1: Application** Having multiple cores can speed up smaller problems but this will not be enough for larger problems. Supercomputers are prohibitively exceptionally expensive to buy

			best. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2-Mid Level (5-8 marks)** The candidate demonstrates reasonable knowledge and understanding of methods of utilising large amounts of computing power; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate makes a reasonable attempt to come to a conclusion showing some recognition of which approaches to provide increasingly larger amounts of computing power are best. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1-Low Level (1-4 marks)** The candidate demonstrates a basic knowledge of methods of utilising large amounts of computing power; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an		and run for all but large organisations. GPUs are becoming a cost efficient way of tackling problems. GPUs tend to have large number of cores so can run on highly paralleliseable problems... …but only where the same instruction is being applied to multiple pieces of data (SIMD) **AO3.3: Evaluation** Increased clock speed is limited to smaller problems. Even doubling the clock speed would only halve the time taken. Parallel processing isn't suited to all problems. Most problems are only partially parallelisable. Writing algorithms for parallel processing is more challenging than GPUs suited to a subset of science/ engineering problems where the same calculation is repeated on multiple data sets.

			unsupported assertion. *The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		

OCR
Oxford Cambridge and RSA

A Level Computer Science

H446/02 Algorithms and programming

Practice paper - Set 1

Time allowed: 2 hours 30 minutes

Do not use:
- a calculator

First name										
Last name										
Centre number						**Candidate number**				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided.
- If additional space is required, use the lined page(s) at the end of this booklet. The question number(s) must be clearly shown.
- Do **not** write in the barcodes.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **28** pages.

Answer **all** the questions.

Section A

1 A flight simulator allows a user to take control of a simulated aeroplane. The user can fly the plane in an environment that can simulate different weather conditions and additional planes in the sky.

(a) Identify **three** pieces of information that would need to be researched in order to design this simulator.

1. .....

.....

2. .....

.....

3. .....

..... **[3]**

(b) Explain what is meant by 'concurrent processing' and describe **one** example of how the simulator could make use of it.

Concurrent processing .....

.....

Example .....

.....

.....

.....

.....

..... **[4]**

(c) Air traffic controllers are considering introducing a new flight path.

Explain **two** reasons why they might use the new flight path in the simulation before implementing it in the real world.

1. .....

.....

.....

2. .....

.....

.....

..... **[4]**

(d) Abstraction has been used in the design and creation of the flight simulator.

Explain, using an example, the need for abstraction in the creation of the flight simulator.

.....

.....

.....

.....

.....

..... **[3]**

2 The layout for a 2-player board game is shown in Fig 2.1

START	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
END	62	61	60	59	58	57	56

Fig 2.1

The game is played by rolling two 6-sided dice and moving that number of spaces. Both players start on the START space. If a player lands on a space occupied by the other player, they move to the next available space.

The board is to be stored as a 2-dimensional array.

(a) The board shown in Fig 2.1 is a visualisation of the problem. Explain what visualisation means in this example.

.....

.....

.....

..... **[2]**

(b) Each time a player moves, a series of obstacles are to be added to the board.

On their turn, each player rolls two dice. The smaller number from the two dice is taken, and that many obstacles will appear on the board in random locations.

For example, if a 3 and 6 are rolled, then 3 obstacles will appear.

A recursive function is written in pseudocode to perform this task.

```
01 function generateObstacle(diceNumber)
02     if diceNumber == 0 then
03         return true
04     else
05         x = randomNumber(0, 7)
06         y = randomNumber(0, 7)
07         board(x, y) = new obstacle()
08         generateObstacle(diceNumber-1)
09     endif
10 endfunction
```

The code `new obstacle()` generates an instance of the object obstacle.

(i) Explain the purpose of the code in line `01` in the algorithm.

.....

.....

.....

..... **[2]**

(ii) Identify the line of code where recursion occurs.

..... **[1]**

(iii) The recursive function could have been written using iteration.

Describe the benefits and drawbacks of using recursion instead of iteration.

Benefits .....

.....

.....

.....

Drawbacks .....

.....

.....

..... **[4]**

(iv) Rewrite the function in part (b) so it uses iteration instead of recursion.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(v) If a position on the board is not occupied, its value is set to a blank string ("").

The current algorithm does not check if the random space generated is currently occupied.

Write a subroutine that takes the generated position of the board, checks if it is free and returns `true` if free, or `false` if occupied.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

(c) The programmer is using a number of subroutines in the program. Explain, using an example, the benefits to the programmer of using subroutines in the creation of this game.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(d) The programmer has been told the recursive function has the Big O notation of O(n).

(i) State the purpose of Big O notation.

.....

..... **[1]**

(ii) Explain what the Big O notation O(n) means for this recursive function.

.....

..... **[1]**

3* A multi-national retailer has a very large database storing customers, stock and orders.

The retailer uses data mining to retrieve a variety of information.

Evaluate the use of data mining by the retailer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

4 A 1-dimensionsal array stores a set of numbered cards from 0 to 7. An example of this data is shown in Fig in 4.1

2	0	1	7	4	3	5	6

Fig 4.1

(a) The programmer wants to search for a specific card in the array.

State whether a binary search or a linear search would be the most appropriate method to search for a specific card, and justify your answer.

Search method.....

Justification .....

.....

.....

..... **[3]**

(b) A programmer is writing a computer program to sort the cards into the correct order (0 to 7).

(i) Show how an insertion sort would sort the array in Fig 4.1 into the correct order. Draw the array after each move.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

(ii) Describe how a quick sort algorithm works with the data in Fig 4.2.

2	0	1	7	4	3	5	6

Fig 4.2

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(c)* Two other sorting algorithms the programmer could have used are a merge sort and bubble sort.

The worst case scenario for Merge is O(n log(n)) and for Bubble is O(n^2).

Compare the use of a merge sort and a bubble sort on this array, evaluating the performance of each sort, making reference to the worse case Big O notation.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[9]**

5 A procedure is shown below.

```
01   procedure fun1(x)
02      y=""
03      if x < 0 then
04         flag = true
05         x = x * -1
06      endif
07      while (x > 0)
08         y = str(x MOD 2) + y
09         x = x DIV 2
10      endwhile
11      if flag == true then
12         y = "1" + y
13      else
14         y = "0" + y
15      endif
16      print(y)
17   endprocedure
```

`flag` is a local variable and has a default value of `false`.

(a) Explain why `str` is needed in line `08`.

.....

.....

.....

.....

..... **[3]**

(b) (i) Show the result of `y` when the procedure is called with: `fun1(10)`. Show your working.

`y`: ..... **[4]**

(ii) Show the result of `y` when the procedure is called with `fun1(-13)`. Show your working.

`y`: ..... **[4]**

(b) (iii) Identify the purpose of this algorithm.

.....

..... **[1]**

(c) In this procedure, `flag` is assumed to be a local variable.

(i) Explain the problem that would be caused in this algorithm if `flag` was a global variable.

.....

.....

.....

.....

.....

..... **[3]**

(ii) The programmer has chosen to keep `flag` as a global variable.

Describe how the algorithm could be changed to prevent the error identified in part (i)

.....

..... **[1]**

6 A salesman travels around the country, stopping at specific places, and then returning to the starting place.

Fig 6.1 shows an example map of places that the salesman visits.

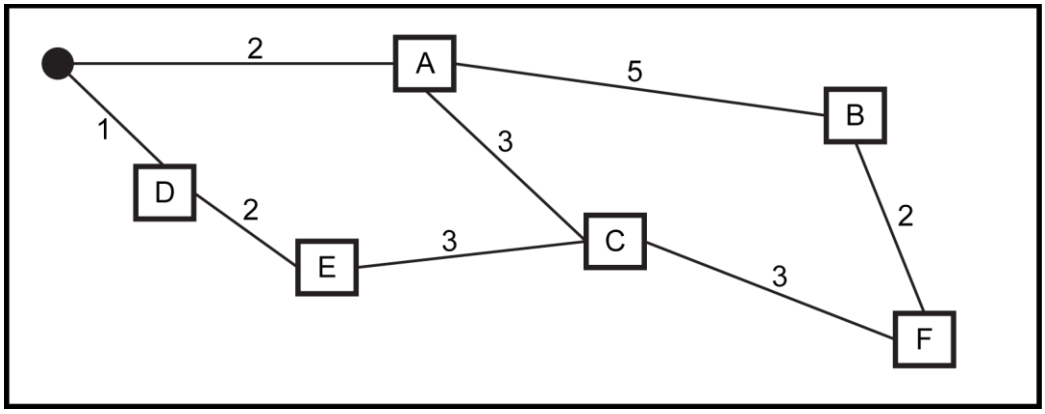


Fig 6.1

The filled in circle represents the start and end point. The letters represent the places to visit. The lines are the routes available and the numbers are the length of time each route takes to travel.

(a) Explain how abstraction has been applied in the production of Fig 6.1

.....

.....

.....

..... **[2]**

(b) The travelling salesman aims to find the shortest route between these places to visit.

A programmer is writing an algorithm to solve the travelling salesman problem.

The programmer is using a tree to find the most efficient route. Fig 6.2 shows part of the tree with three levels completed.

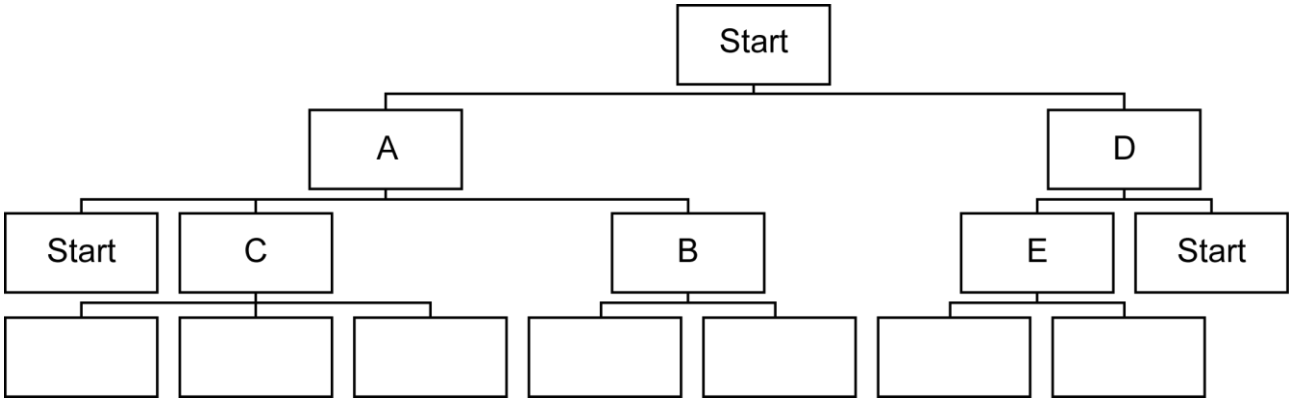


Fig 6.2

(i) The 'Start' nodes on level three are not expanded again as this is a repeat, 'Start' has already been expanded.

Write the place names in the boxes in Fig 6.2, to complete the fourth level of the tree structure for the map shown in Fig 6.1.

[3]

(ii) Explain why the tree in Fig 6.2 is **not** a binary tree.

.....

..... **[1]**

(c) The programmer has decided to use a graph instead of a tree structure.

(i) Describe what is meant by a graph structure.

.....

.....

.....

..... **[2]**

(ii) The pseudocode below shows part of an algorithm which uses a queue to traverse the graph breadth-first. Complete the missing elements of the algorithm.

```
markAllVertices (notVisited)
createQueue()
start = ..............
markAsVisited(..............)
pushIntoQueue(start)
while QueueIsEmpty() == ..............
    currentNode = removeFromQueue()
    while allNodesVisited() == false
        markAsVisited(..............)
        //following sub-routine pushes all nodes connected to
        //currentNode AND that are unvisited
        pushUnvisitedAdjacents()
    endwhile
endwhile
```

[4]

(d) Fig 6.3 is a graph representation of the places that the travelling salesman visits. Using this graph, show how Dijkstra's algorithm would find the shortest path from place A to place F.

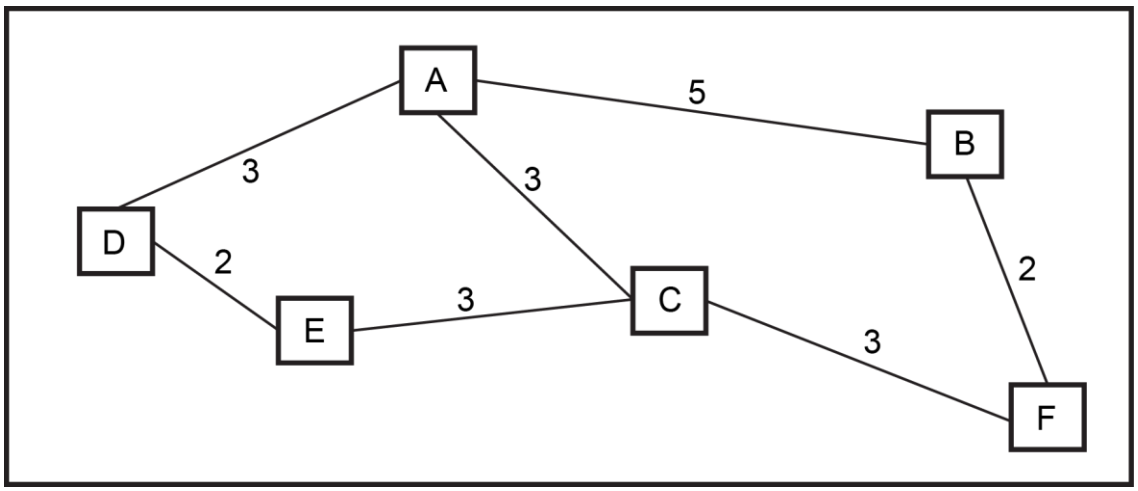


Fig 6.3

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

Section B

Answer **all** questions

7 Four in a Row is a game where two players drop coloured discs into a grid, with the aim to get four of their own colour in a row.
Each player is given a set of coloured discs, red (R) or yellow (Y). The players take it in turns to drop their disc into a column in the grid. The disc drops down to the lowest available space in that column.

The grids below (Fig 7.1 and 7.2) show what happens when the yellow player drops a disc into column 2:

Before

	0	1	2	3	4	5	6
0							
1							
2							
3		R	Y	Y			
4		Y	R	R	Y		
5	R	Y	R	R	Y	R	

Fig 7.1

After

	0	1	2	3	4	5	6
0							
1							
2			Y				
3		R	Y	Y			
4		Y	R	R	Y		
5	R	Y	R	R	Y	R	

Fig 7.2

The game continues until one player has got four discs of their colour in a straight row in any direction i.e. vertical, horizontal, or a diagonal.

(a) A programmer is going to use decomposition to help produce the game.

(i) Explain how decomposition can be used in the design of the game Four in a Row.

.....

.....

.....

..... **[2]**

(ii) The program will allow the players to take it in turns to make a move. Each move will be checked to ensure it is valid (i.e. the column is not already full). After each move the program will check if that player has won by checking the horizontal, vertical and diagonal positions to confirm if that player has four discs in a row.

The programmer has developed a top-down design for the program as shown in the structure diagram Fig 7.3.

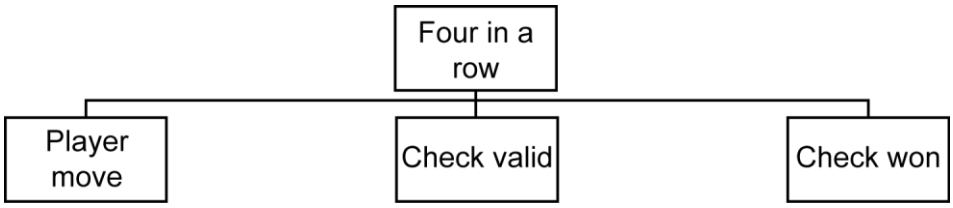


Fig 7.3

Add **one** further level to the structure diagram, by dividing the sub-modules 'Player move', 'Check valid' and 'Check won' into further sub-modules.

[3]

(a) (iii) The structured design for this program makes use of pipelining. Describe **one** example of where pipelining could be used in this program.

.....

.....

.....

..... **[2]**

(b) A 2-dimensional array, `grid`, is used to hold the game grid.

Using pseudocode, write a function that takes as input the player whose turn it is, and the column number they select as their turn. The function either:

- returns 999 (i.e. the column is already full), or
- stores the player's move in the array and returns the row the disc has been placed in.

Annotate your pseudocode with comments to show how it solves the problem.

[6]

(c) After a player makes their move, the program needs to check if that player has won (i.e. the player has four discs in a row).

Subroutines have already been written to check if the player has won vertically, or diagonally.

Using pseudocode, write a procedure that reads appropriate parameters and checks if the player has won horizontally. If the player has won, display an appropriate message identifying which player has won.

[6]

(d) (i)* The programmer is writing a new version of the game, where each player removes one disc from the bottom row of the grid before a new move is made.

In the example below, player R removes one disc from column 2 (Before) and places one in column 4 (After).

Before

	0	1	2	3	4	5	6
0							
1							
2							
3		R	Y	Y			
4		Y	R	R	Y		
5	R	Y	R	R	Y	R	

Fig 7.4

After

	0	1	2	3	4	5	6
0							
1							
2							
3		R		Y	R		
4		Y	Y	R	Y		
5	R	Y	R	R	Y	R	

Fig 7.5

The programmer has to decide whether to continue to use a 2D array, or produce an array of queues.

Evaluate the use of a 2D array versus an array of queues to perform this action.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

(d) (ii) Explain why a stack would **not** be an appropriate data structure for this revised game.

.....

.....

.....

..... **[2]**

(d) (iii) A procedure needs to be written to remove the disc from the chosen column. The procedure will:

- have the column the disc is being removed from as a parameter
- move each disc in that column down to the bottom of the grid
- replace the top space with an empty string ("")

Complete the algorithm below.

```
procedure playDisc (removeColumn)
```

[3]

(e) The programmer is adding a feature that allows a player to play against the computer.

Explain how the programmer could make use of a tree structure, using a heuristic approach in the development of the computer player.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[7]**

END OF QUESTION PAPER

ADDITIONAL ANSWER SPACE

If additional answer space is required, you should use the following lined page(s). The question number(s) must be clearly shown in the margin(s).

OCR
Oxford Cambridge and RSA

Practice Paper 1

GCE Computer Science

H446/02 Algorithms and Programming

Duration: 2 hour/s 30 minutes

MAXIMUM MARK 140

This document consists of 34 pages

MARK SCHEME:

Question			Answer	Marks	Guidance
1	a		1 mark per data item, accept any appropriate, sensible suggestions • Number of other planes that could be in the sky (1) • Speed(1) • Flight path(1) • Altitudes(1) • Rate of acceleration(1)	**3** **AO2.2 (3)**	
1	b		Max 1 for explanation of concurrent programming. Max 3 for each example. Concurrent processing: • One process does not have to finish before the other starts(1) Example e.g. • Each plane can move independently(1) • All move at the same time (1) • All need to react to different events(1) • The weather(1) • Wind, rain, direction of air etc. (1) • Each element needs to be run simultaneously(1) • It will react to its own stimulii(1)	**4** **AO1.2 (1)** **AO2.1 (3)**	Accept any reasonable suggestion for concurrent programming in the simulator For examples: 1 mark for identifying example. 1 mark for saying how they act concurrently. 1 mark for saying why this is necessary.
1	c		1 mark per bullet e.g. • It is safer... (1) • ...Real planes/lives are not put at risk by testing it in reality(1) • Time can be sped up/decreased ... (1) • ... do not need to wait to see what happens, can view changes immediately(1) • It will cost less... (1) • ...Can make multiple changes/test all possibilities(1)	**4** **AO1.1 (2)** **AO2.1 (2)**	

Question			Answer	Marks	Guidance
1	d		1 mark per bullet to max 3 • Removing unneeded complexities (1) • Saves memory/resources (1) • E.g. remove passengers, other planes, other obstacles, landscaping to reduce memory needed (1)	**3** **AO1.2 (2)** **AO2.2 (1)**	
2	a		2 marks, 1 for defining visualisation, 1 for application to the 2-d array and grid • Presents data in an easy-to-grasp way(1) • An array is not actually a grid/table(1)	**2** **AO1.1 (1)** **AO2.1 (1)**	
2	b	i	1 mark per bullet to max 2 • Declares a function called `generateobstacle`(1) • Has parameter `diceNumber` (1)	**2** **AO1.1 (1)** **AO2.1 (1)**	
2	b	ii	• 08(1)	**1** **AO2.1 (1)**	
2	b	iii	max 3 marks for benefit, max 3 for drawback, max 4 marks overall Benefit • More natural to read (1) • Quicker to write/less lines of code.. (1) As some functions are naturally recursive(1) • Suited to certain problems (1) For example those using trees (1) • Can reduce the size of a problem with each call.(1) Drawback • Can run out of stack space/memory(1) (due to too many calls (1)) causing it to crash(1) This can be avoided with tail recursion (1) • More difficult to trace/follow(1) as each frame on the stack has its own set of variables(1) • Requires more memory than the equivalent iterative algorithm. • Usually slower than iterative methods (1) due to maintainence of the stack (1)	**4** **AO1.1 (2)** **AO1.2 (2)**	

Question			Answer	Marks	Guidance

2	b	iv	1 mark per bullet
			• Loop start and end in correct positions(1)
			• With correct number of iterations(1)
			• Returns a value(1)
			• All other code correct, in the right place(1)
			
			e.g.
			`function generateobstacle(diceNumber)`
			`  for count = 0 to diceNumber`
			`  x = randomNumber(0, 7)`
			`  y = randomNumber(0, 7)`
			`  board(x, y) = new obstacle()`
			`  next count`
			`  return true`
			`endfunction`
				4
				AO2.2 (1)
				AO3.2 (3)	

2	b	v	1 mark per bullet, to max 3
			• Appropriate declaration of function, taking 2 parameters(1)
			• Checks position in board against "" correctly(1)
			• Returns `false` and true correctly(1)
			e.g.
			`function checkFree(x, y)`
			`  if board(x, y) == "" then`
			`  return true`
			`  else`
			`  return false`
			`  endif`
			`endfunction`
				3
				AO2.1 (1)
				AO3.2 (2)	

2	c		Max 2 marks for explanation of benefits. Max 2 marks for example related to this scenario • Code can be re-used(1) • …Saves time (1) • Can use subroutine(s) in other programs(1) • …saves time(1) • Can test independently…(1) • … may make finding errors easier(1) • Any suitable example, e.g. the code for rolling dice can be written once (1), then called whenever needed in the game (1)	**4** **AO1.1 (1)** **AO1.2 (1)** **AO2.1 (2)**	The question states there is only 1 programmer, so splitting the code and giving it to different programmers is not relevant to this scenario
2	d	i	1 from • Evaluate the complexity of the algorithm(1) • Show how the time/memory/resources increase as the data size increases (1) • Evaluate worst case scenario for the algorithm (1)	**1** **AO1.1 (1)**	
2	d	ii	• As the dice no increases, the time the function takes to run increases proportionally / linearly (1)	**1** **AO2.1 (1)**	
3			**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of data; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context proided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of data mining; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Data mining looks through vast quantities of data • Searches for relationships between facts/components/events that may not be obvious • May include pattern matching algorithms • May involve anomaly detection algorithms • Used for business modelling • Used to plan for future

			The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of data mining with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		eventualities **AO2: Application** • Can look for links between a customer's purchases • Give recommendations for future purchases • Check for days/times/months where increases are likely and what the increase will be purchasing • Look at matching sales, when people buy one product what else do they buy with it **AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of using data mining. e.g. • Can improve marketing • Can improve quantity of stock needed • Ensure demand is met • Increase sales/profit • Takes vast processing requirements • Need powerful computers • Privacy concerns from customers • Misuse of information • Inaccurate information

					can produce false results
4	a		1 mark for linear search, 2 for justification Justification: • The array is not sorted(1) • Linear does not need ordered/linear goes through all elements from beginning/binary needs a sorted array(1)	**3** **AO1.1 (2)** **AO2.1 (1)**	
4	b	i	1 mark for each set of 2 moves • 02174356 01274356(1) • 01247356 01234756(1) • 01234576 • 01234567(1)	**3** **AO1.1 (2)** **AO2.1 (1)**	Allow follow through if one move is incorrect

<table>
<tr>
<td>4</td>
<td>b</td>
<td>ii</td>
<td>1 mark per bullet to max 6

Uses divide-and-conquer(1)
First item becomes pivot / 2 is the pivot(1)
Compare each item to the pivot (e.g. compare 0 to 2, then 1 to 2)(1)
Make two lists, 1 with less than the pivot... (0, 1) (1)
... 1 with more than the pivot (7,4,3,5,6) (1)
Quick sort the new lists(1)
Recombine the sub-lists(1)

OR
Example of alternative quicksort method
<table>
<tr><td></td><td>2 →</td><td>0</td><td>1</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0 →</td><td>1</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1 →</td><td>7</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>7 →</td><td>4</td><td>3</td><td>5</td><td>6 ←</td></tr>
<tr><td>Swap 7 and 6</td><td>2</td><td>0</td><td>1</td><td>6 →</td><td>4</td><td>3</td><td>5</td><td>7 ←</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>6 →</td><td>4</td><td>3</td><td>5 ←</td><td>7</td></tr>
<tr><td>Swap 5 and 6</td><td>2</td><td>0</td><td>1</td><td>5 →</td><td>4</td><td>3</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4 →</td><td>3</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4</td><td>3 →</td><td>6 ←</td><td>7</td></tr>
<tr><td></td><td>2</td><td>0</td><td>1</td><td>5</td><td>4</td><td>3</td><td>6 ← →</td><td>7</td></tr>
</table>
marks for:

Uses divide-and-conquer(1)
Highlight first list element as start pointer, and last list element as end

</td>
<td>6
AO1.1 (1)
AO1.2 (2)
AO2.1 (3)</td>
<td>If no description i.e. the candidate has just shown the quick sort, max 4 marks.</td>
</tr>
</table>

			pointer • Repeatedly compare numbers being pointed to... • ...if incorrect, swap and move end pointer • ...else move start pointer • Split list into 2 sublists • Quick sort each sublist • Repeat until all sublists have only 1 number • Combine sublists		

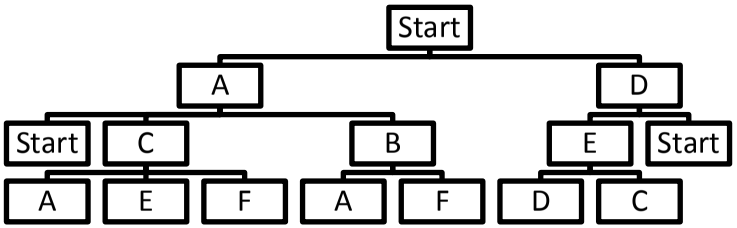
4	c		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of merge and bubble sorts; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of merge and bubble sorts; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of merge and bubble sorts with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks**	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Merge sort uses sub-lists • Bubble sort uses a temp element • Bubble sort moves through the list repeatedly • Merge sort divides the list into smaller lists • Merge is a recursive algorithm • Worst case is logarithmic, scales up well • Worst case is exponential, does not scale up well **AO2: Application** • Small data set • Few changes are needed • Demonstrates use of merge and/or bubble on the array • Calculations of average speed/best speed/worse speed **AO3: Evaluation** Candidates will need to evaluate the benefits and

			No attempt to answer the question or response is not worthy of credit.		drawbacks of each sorting algorithm e.g. • Merge is fast on large data sets • Bubble is intuitive (easier to program) • Both are fast (or even) on smaller data sets • Bubble's average speed is worse than merge • Bubble will be easier to write for such a small data set • Accept argument for either way as long as justified
5	a		max 1 mark for purpose and max 2 for application to line 8. • Case/convert numeric value to a string • Result of `x MOD 2` will be a number... • ...needs to be concatenated to string y	**3** **AO1.2 (1)** **AO2.2 (2)**	**Accept "avoid type mismatch error" for 1 mark as part of application to line 8**
5	b	i	1 mark for `flag = false`. 1 mark for showing y's values correctly through loop 1 mark for showing x's values correctly through loop	**4** **AO1.2 (1)** **AO2.1 (3)**	If only the result is shown, 1 mark only. Award bod if no "" with y

			1 mark for the correct answer
			
			• `10 > 0` (1)
			• `y = "0"`
			`x = 5`
			`y = "10"`
			`x = 2`
			`y = "010"`
			`x = 1`
			`y = "1010"` (1 for y at each stage)
			`x=0` (1 for x at each stage)
			• `01010` (or `y = "01010"`)
					Accept any suitable answer, e.g. trace table

5	b	ii	1 mark for `flag = true` and `x=13`
			1 mark for showing y's values correctly through loop
			1 mark for showing x's values correctly through loop
			1 mark for the correct answer
			
			• `flag = true`
			`x = 13`
			• `y = "1"`
			`x = 6`
			`y="01"`
			`x=3`
			`y="101"`
			`x=1`
			`y="1101"` (1 for y at each stage)
			`x=0` (1 for x at each stage)
			• `result=11101 (or y="11101")`
				4
				AO1.2 (1)
				AO2.1 (3)
					If only the result is shown, 1 mark only.
					Award bod if no "" with y
					
					Accept any suitable answer, e.g. trace table

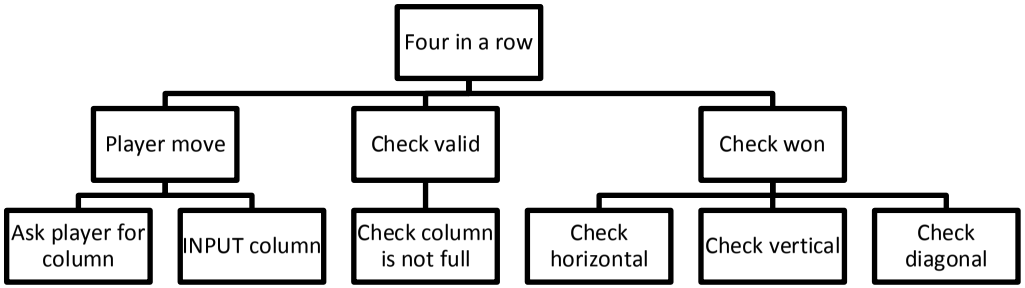
5	b	iii	• Converts a denary number into sign and magnitude	**1** **AO2.1 (1)**	cao
5	c	i	1 mark per bullet to max 3 • Past values will remain(1) • If the last value was a negative number it will remain `true`... (1)	**3** **AO1.2 (1)** **AO2.1 (2)**	

			• ...future positive values will still have the flag `true` / will be treated as a negative number(1)		
5	c	ii	1 from • Put an else statement in/another IF to set flag to `false` if needed(1) • Make `flag` be `false` in the first line of the procedure(1)	**1** **AO2.2 (1)**	
6	a		1 mark per bullet to max 2 e.g. • Places have been replaced with variables ... (1) • ...e.g. a place has been replaced with A(1) • Irrelevant information has been removed... (1) • ... e.g. only the routes and places are shown(1) • Time is given as a numeric value(1)... • ...e.g. 1 rather than 1 hour, or 1 minute(1) • Relative geographic location may not be accurate (1) • ... e.g. positions of the towns may not be proportional to actual distance (1)	**2** **AO1.2 (2)**	
	b	i	1 mark for completing A,E,F below C 1 mark for completing A, F below B 1 mark for completed D, C below E 	**3** **AO2.2 (3)**	

6	b	ii	In a binary tree a node can only have two children	**1** **AO1.2 (1)**	
6	c	i	1 mark per bullet to max 2 • Collection of data nodes/vertices(1) • Connections/edges are set between nodes/vertices(1) • Graph (edges) can be directional or bi-directional(1) • Graphs (edges) can be directed or undirected(1)	**2** **AO1.1 (2)**	

6	c	ii	1 mark each
			
			`markAllVertices (notVisited)`
			
			`createQueue()`
			
			`start = currentNode  (1)`
			
			`markAsVisited(start)  (1)`
			
			`pushIntoQueue(start)`
			
			`while QueueIsEmpty() == false  (1)`
			
			`  popFromQueue(currentNode)`
			
			`  while allNodesVisited() == false`
			
			`  markAsVisited(currentNode)  (1)`
			
			`  //following sub-routine pushes all nodes connected to`
			`  //currentNode AND that are unvisited`
			`  pushUnvisitedAdjacents()`
			
			`  endwhile`
			
			`endwhile`
				4
				AO1.2 (2)
				AO2.1 (1)
				AO3.2 (1)	

6	d		Max 6. 1 mark for final solution, max 5 for showing the stages • Mark A as the current node(1) • Record B is 5, C is 3, D is 3(1) • Mark A as visited(1) • C is shortest distance from A(1) • (C as current) Record E as 6, F as 6(1) • Mark C as visited(1) • (D as current) Record E as 5(1) • Mark D as visited(1) • (B as current) Record F as 7, do not update table as longer(1) • Mark B as visited(1) • (E as current) Record D as 8, do not update table as longer and E as visited(1) • A-C-F found as shortest(1)	**6** **AO1.2 (3)** **AO2.1 (3)**	
7	a	i	1 mark per bullet to max 2 • Breaks a problem down into its component parts(1) • Game can be divided into subprograms(1) • Subprograms can then be programmed as subroutines(1)	**2** **AO1.1 (2)**	

7	a	ii	1 mark for identifying all of the sub-modules in each sub-module. Allow any appropriate additions, do not award a mark for any inappropriate or clearly incorrect sub-modules. • Player move must ask for the move to be input(1) • Check valid must check if the column is full(1) • Check won must check horizonal, vertical and diagonal(1) e.g.  	**3** **AO2.2 (3)**	
7	a	iii	1 mark for the example, 1 mark for explanation of why it is pipelining e.g. • The results from the player move subroutine ... (1) • ...will feed into the check valid subroutine(1)	**2** **AO1.2 (1)** **AO2.1 (1)**	Accept any valid example

7	b		Programming steps to award marks for, max 6
			• Function declaration taking 2 parameters(1)
			• Looping through all 6 elements in the array…(1)
			• …in the correct order (bottom to top, 5 to 0) (1)
			• Place player in correct position…(1)
			• … return the position(1)
			• Return 999 if no space in that column(1)
			
			Example pseudocode
			e.g.
			`function gameMove(player, column)`
			`  for x = 5 to 0 step -1`
			`  if grid(x,column) == "" then`
			`  grid(x,column) = player`
			`  return x`
			`  endif`
			`  next x`
			`  return 999`
			`endfunction`
				6
				AO2.2 (2)
				AO3.2 (4)	

7	c		The algorithm can be tackled by either a) Checking all possible positions, b) Just checking what is around the last move c) checking the entire row. Full marks can be awarded for all possible methods, if correct.
			
			Programming steps to be awarded as follows, to max 6 marks.
			• Appropriate procedure declaration…
			• … taking at least player parameter(1)
			• Checking each element in the row(1)
			• Only checking valid options (e.g. if checking row-3, row-2, row-1, 0 then they need to check all rows are within the grid) (1)
			• Updating a counter/checking for four-in-a-row(1)
			• Appropriate output message(1)
			
			Example pseudocode:
			
			`procedure checkHorizontal (player, row)`
			
			`counter = 0`
			`for x = 0 to 6`
			`if grid(row, x)== player then`
			`counter = counter + 1`
			`if counter >= 4 then`
			`print "Player " + player + " has won"`
			`endif`
			`else`
			`counter = 0`
			`endif`
			`next x`
			
			`endprocedure`
				6
				AO2.2 (2)
				AO3.2 (4)	

7	d	i	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of queues and arrays; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of queues and arrays; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of queues and arrays with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks**	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Arrays are static (size cannot change) • Queues are dynamic (size can change) • Queues use pointers to identify the first element (to be removed) **AO2: Application** • Array will need all elements to be moved 'down 1' each time a disc is removed • Queue will allow the front element to be taken out and then the pointer will move • Algorithms for queues can be more complex, especially as the language may use an array to implement the queue **AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of using queues and arrays and suggest an appropriate solution e.g. • Size does not need to

			No attempt to answer the question or response is not worthy of credit.		change (Static is needed as grid is fixed size) so that benefit of queues is not necessary • Programmer has already written a program using arrays, may be less time consuming to edit it for arrays • Language may need a queue to be programmed in an array, therefore an array may be more straight forward to use • Queue does not need to move all elements each time a counter is removed, only pointers change
7	d	ii	Max 2 • Stack is last-in-first-out(1) • This game the first-in needs to be first-out(1)	**2** **AO1.1 (1)** **AO2.1 (1)**	

7	d	iii	1 mark per bullet to max 3
			• Appropriate loop to move through the column (bottom to top, so 5 to 1)
			• Replacing the grid with the value above (5 with 4 etc.)
			• Replacing row 0 with ""
			e.g.
			`procedure playDisc (removeColumn)`
			`  for x = 5 to 1 step -1`
			`  grid(x, removeColumn) = grid(x-1, removeColumn)`
			`  next x`
			`  grid(0, removeColumn) = ""`
			`endprocedure`
				3
				AO2.2 (1)
				AO3.2 (2)	

7	e		1 mark per bullet to max 7 • Stores/considers a range of next moves(1)	**7** **AO1.2 (2)**	Allow a diagrammatic answer with appropriate annotation

			• Creates branches with possible further moves (from both players) (1) • Continues branching(1) • Ranks possible moves based on success down the branches(1) • Uses a branching algorithm/step to decide which direction to follow(1) • Can increase ranks on stored moves based on past moves(1) • The current board position is at the top of the tree (1) • The possible moves are at the next level (1) • Each level gives every possible next move (1) • Can add weightings as to probability of winning (1) • Searching algorithm can find which set of moves leads to/has greatest possibility of winning (1) • Heuristics can help in the actual ranking / probability generation for each game state. (1)	**AO2.1 (2)** **AO2.2 (3)**	that meets the bullets

OCR
Oxford Cambridge and RSA

A Level Computer Science

H446/02 Algorithms and programming

Practice paper – Set 2

Time allowed: 2 hours 30 minutes

Do not use:
- a calculator

First name										
Last name										
Centre number						Candidate number				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided. Additional paper may be used if required but you must clearly show your candidate number, centre number and question number(s).
- Do **not** write in the barcodes.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **28** pages.

Answer **all** the questions.

Section A

1 A binary search tree, `colour`, stores data about colours that are entered into a computer.

(a) A binary search tree is one example of a type of tree.

(i) State the main features of a tree.

.....

.....

.....

.....

.....

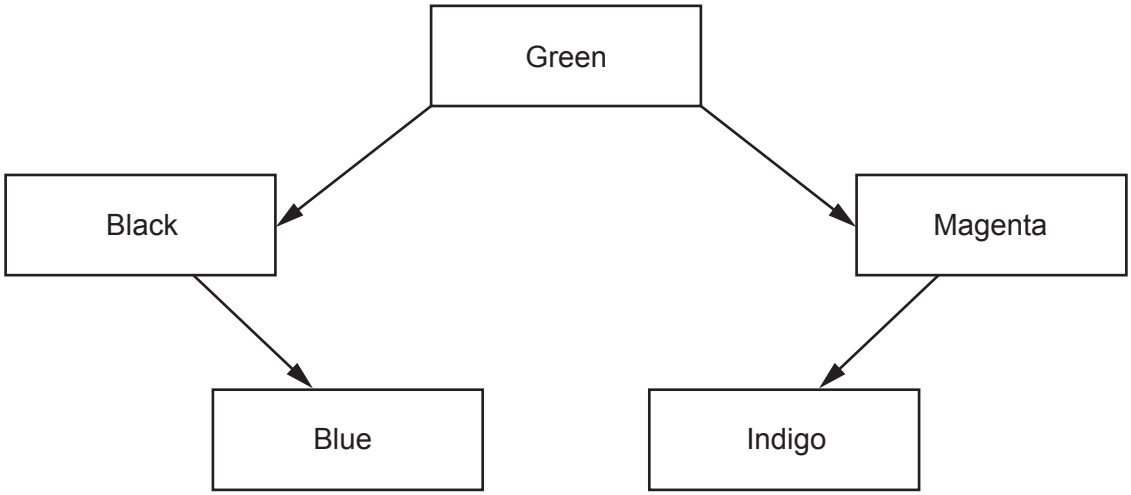
..... **[3]**

(ii) State the features that make a tree a binary search tree.

.....

..... **[1]**

(b) The current contents of `colour` are shown.

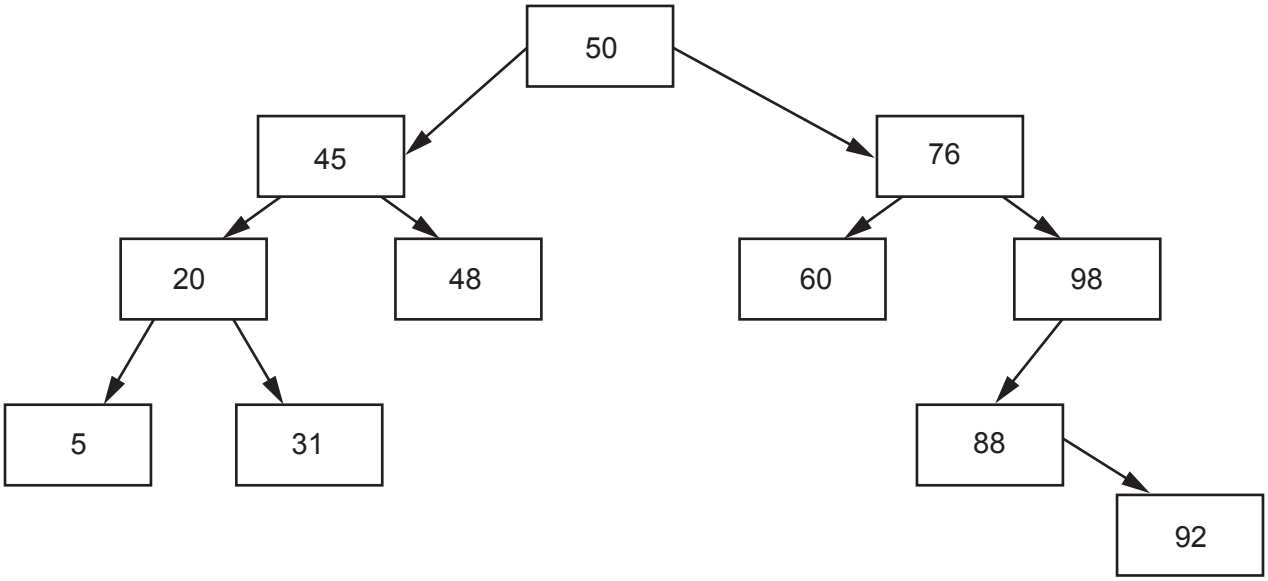


Add the following colours to the tree above in the order written:

Brown White Orange Purple

[4]

(c) A second binary search tree, `numbers`, stores numbers that are entered into a computer. The contents of the tree are shown below:



(i) Explain, using the binary search tree `numbers` as an example, how a depth-first (post-order) traversal is performed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(ii) Explain, using the binary search tree numbers as an example, how a breadth-first traversal is performed.

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(d) The binary search tree, `values`, is stored as an array of nodes. Each node has a left pointer, a right pointer and the data being stored. The following data is entered in the order shown below:

68 30 73 22 1 90 70

The following table shows the data stored in the array. The Root Pointer stores the node number of the first element in the tree.

Root Pointer

0

Free Pointer

7

Array Index	Left Pointer	Data	Right Pointer
0	1	68	2
1		30	
2		73	
3		22	
4		1	
5		90	
6		70	

Table 1.1

(i) Complete the remaining Left Pointer and Right Pointer values for the data entered in Table 1.1. Where the pointer is null, leave the space empty. **[3]**

(ii) State the purpose of the Free Pointer.

.....

..... **[1]**

(iii) The following data is added to the array in the given order:

6 100

Add the new nodes to Table 1.1 and update any relevant pointers. **[4]**

BLANK PAGE

2 Fig. 2.1 shows the flight paths between a country's airports. The value in bold beneath each node is the heuristic value from E.

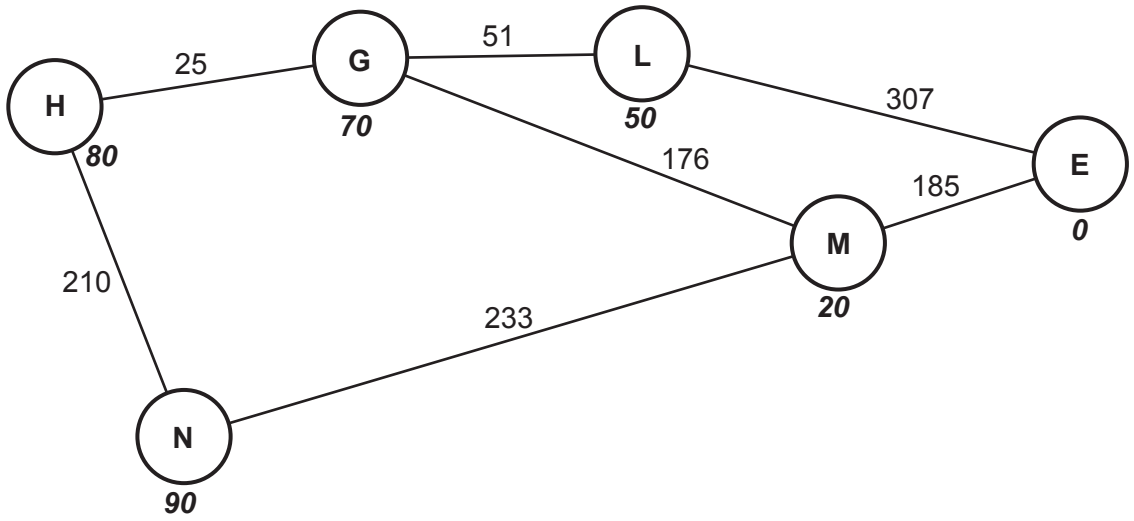


Fig. 2.1

(a) State the full name of the data structure shown in Fig. 2.1.

..... **[2]**

(b) The structure in Fig. 2.1 is searched using the A* algorithm making use of the heuristic values.

(i) State what the heuristic values could represent in Fig. 2.1.

.....

..... **[1]**

(ii) State the purpose of heuristic values in the A* algorithm.

.....

..... **[1]**

(iii) Perform an A* algorithm on the data structure in Fig. 2.1 to find the shortest distance between H and E. Show each step of the process, and the calculations performed for each node visited.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[8]**

(iv) Give **one** decision that is made in the A* algorithm, and describe the effect of this decision on the next step(s) of the algorithm.

Decision .....

.....

Effect .....

.....

.....

.....

..... **[3]**

(c)* A programmer is interested in using concurrent processing to perform a searching algorithm.

Explain how concurrent processing could be used in searching algorithms, and evaluate the benefits and trade-offs from implementing concurrent processing in a searching algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

3 Dexter is leading a programming team who are creating a computer program that will simulate an accident and emergency room to train hospital staff.

(a) Identify **two** features of the problem that make it solvable by computational methods.

.....

.....

.....

.....

..... **[2]**

(b)* Dexter has used decomposition and abstraction during the analysis of the problem.

Explain and evaluate the use of decomposition and abstraction in the creation of this simulation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

(c) Dexter has been told he should make use of caching in the simulation.

Describe what is meant by caching and explain how caching can be used within the simulation.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(d) Two of Dexter's programmers have developed different solutions to one part of the problem. Table 3.1 shows the Big O time complexity for each solution, where *n* = the number of data items.

	Solution A	**Solution B**
Time	O(n)	O(n)
Space	$O(k^n)$ (where k > 1)	O(log n)

Table 3.1

(i) The Big O time complexity for time of each solution is O(n).

Explain what is meant by time complexity, with reference to the solutions' Big O time complexity.

.....

.....

.....

.....

.....

..... **[3]**

(ii) Name the space complexity for each solution:

Solution A .....

Solution B .....

[2]

(iii) Explain, with reference to the Big O complexities of each solution, which solution you would suggest Dexter chooses.

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(e) Dexter's team is using an integrated development environment (IDE).

Describe how the programmers could make use of the following IDE tools:

Breakpoints .....

.....

.....

.....

Stepping .....

.....

.....

.....

[4]

4 A program needs to sort an array of lowercase strings into descending alphabetic order. An example of the data is shown in Fig. 4.1.

sheep	rabbit	dog	fox	cow	horse	cat	deer

Fig. 4.1

(a) Show how a bubble sort would sort the data in Fig. 4.1.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(b) The algorithm will make use of a function, `contains`, that compares two strings and checks if the second string contains the first string. For example, calling the function with (`"fox"`, `"foxhound"`) this would return `true`.

The function needs to:

- Take two strings as parameters, `string1` and `string2`
- return `true` if `string1` is contained within `string2`, or both strings are identical
- return `false` if `string1` is not contained within `string2`

Write, using pseudocode, the function `contains`.

Annotate your pseudocode with comments to show how it solves the problem.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[7]**

(c) **(i)** A merge sort could have been used instead of a bubble sort.

Describe how a merge sort differs from a bubble sort.

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

(ii) Name **two** sorting algorithms, other than a bubble sort and merge sort.

1 .....

2 .....

[2]

(d) Show how a binary search would be performed on the array shown in Fig. 4.2 to find the value 'duck'.

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck

Fig. 4.2

.....

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

Section B

5 Kim is writing an object-oriented program for a four player board game. The board has 26 squares that players move around, as shown in Fig. 5.1.

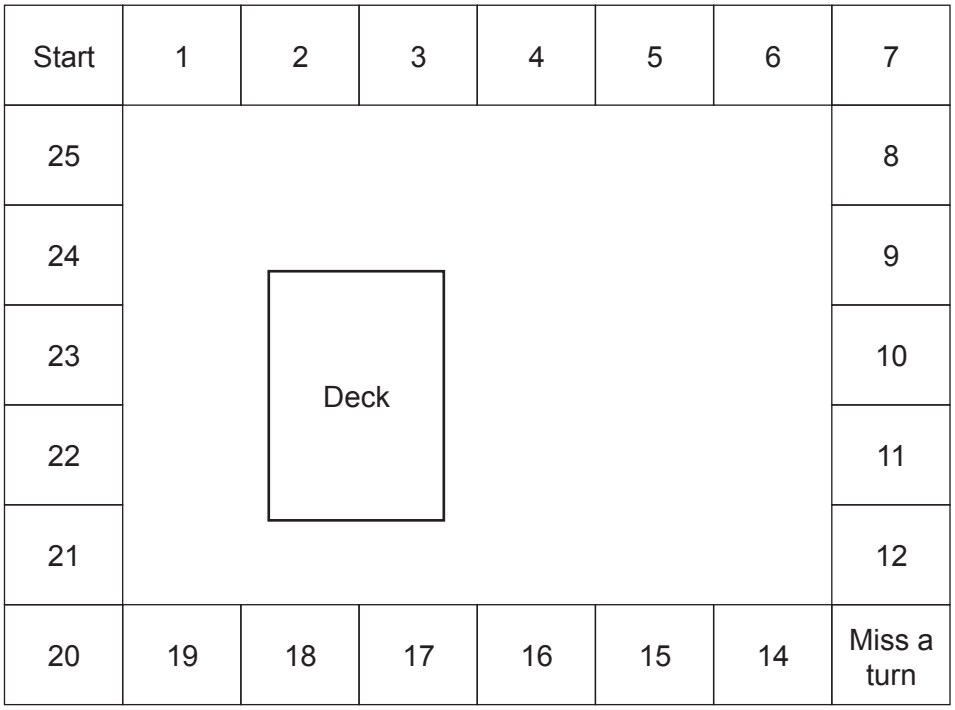


Fig. 5.1

Each player takes it in turn to roll two dice. They then move that number of spaces on the board. If they roll a double (both dice have the same value), they then take a card from the deck. The deck contains 40 cards that each include a sentence (such as “You have won the lottery”). The sentence on the card determines if money is given or taken away from the player.

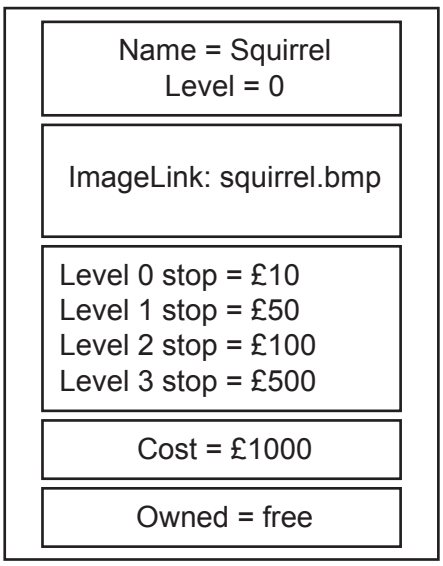


Fig. 5.2

Each square (apart from Start and Miss a turn) has an animal associated with it that the player can purchase, if it has not been purchased already, for example square 6 has a Squirrel. Fig. 5.2 shows an example of one of these animals. Once a player has purchased the animal, any opposing player which subsequently lands on the square/animal has to pay a fine.

Each animal can be upgraded, with each upgrade the game charges more each time a player stops on them. For example, with no upgrade the level 0 squirrel costs £10 when a player stops on it. If £1000 is paid to upgrade, the squirrel is then a level 1 animal and now charges £50 for a stop.

The cost to purchase and upgrade the animal is the same.

Each animal can be upgraded to a maximum of level 3.

When a player lands on, or passes the square ‘Start’ (position 0), they receive £500. If they land on ‘Miss a turn’ (position 13), they miss their next turn.

(a) (i) A class, `Player`, stores the player's ID (P1, P2, P3, P4), their current board position and the amount of money they have.

Fig. 5.3 shows a class diagram for `Player`. A class diagram describes a class. It contains the class name, followed by the attributes, then the methods.

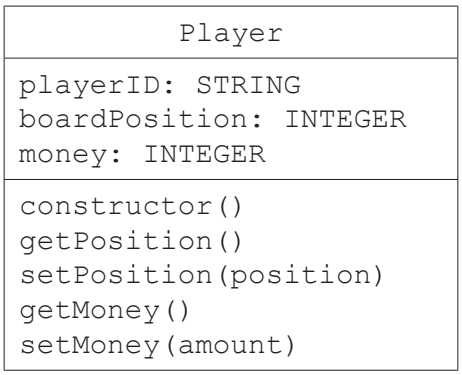


Fig. 5.3

The constructor creates a new instance of `Player`, taking the player's ID as a parameter. The board position is set to 0, and money to £2000.

Write, using pseudocode, the constructor method for the `Player` class.

.....

.....

.....

.....

..... **[3]**

(ii) A class, `Animal`, define the attributes and methods for the animals stored in each square.

Fig. 5.4 shows a class diagram for `Animal`.

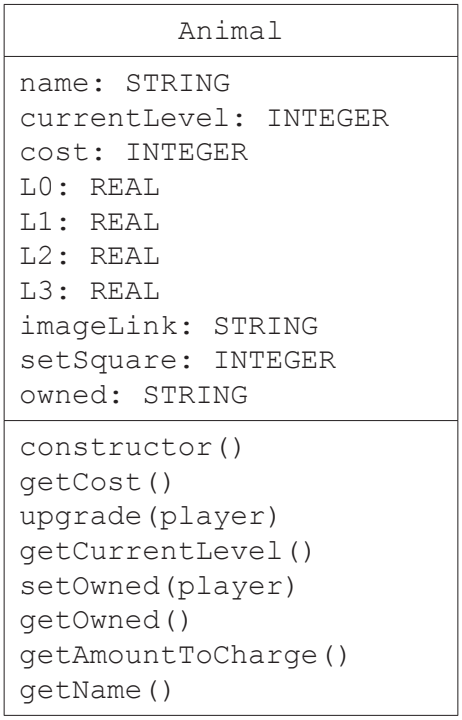


Fig. 5.4

The constructor takes the required data as parameters and then sets `currentLevel` to 0, and assigns the parameters as the remaining attributes for the new object.

Write, using pseudocode, the constructor method for the `Animal` class. **[4]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(iii) Write, using pseudocode, the code to create an instance of `Animal` for the Squirrel shown in Fig. 5.2, positioned on square number 6, for the constructor function you wrote in part **(a)(ii)**.

.....

.....

..... **[2]**

(b) The board is stored as a 1D array, `board`, of data type `Animal`. The spaces at 0, and 13, are left as empty elements that are checked using separate functions.

(i) Complete, using pseudocode, the function to:

- Roll both dice
- Move the player, the dice number of spaces
- If a double is rolled, calls the procedure `pickDeck`
- Adds £500 if they have passed or landed on Start
- Calls the procedure `missAGo` if they land on space 13 or
- Calls the procedure `checkAnimal`
- Return the new position

```
function playerMove(currentPlayer)

    dice1 = random(1,6)

    dice2 = random(1,6)

    position = ..........................  + dice1 + dice2

    if ..............  == dice2 then

        pickDeck(currentPlayer)

    endif

    if position > 25 then

        currentPlayer.setMoney(currentPlayer.getMoney() +  .............)

        position = position -  ..............................

    endif

    if position == ............................... then

        missAGo(currentPlayer)

    elseif position != 0 then

        checkAnimal(currentPlayer)

    endif

    ..............................................................

endfunction
```

[6]

(ii)* The parameter `currentPlayer` from part **(b)(i)** can be passed by value or by reference.

Explain the difference, benefits and drawbacks between passing by value and by reference. Recommend which should be used for `currentPlayer`, justifying your decision.

.... **[9]**

(c) The deck is stored as a zero-indexed 1D array, named `deck`, of type `Card`.

The class diagram for `Card` is shown in Fig. 5.5.

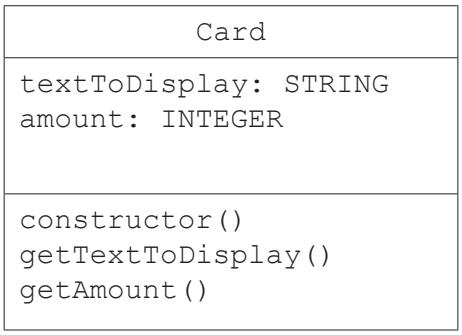


Fig. 5.5

The array, `deck`, is treated as a queue, with a variable, `headPointer`, identifying the first card in the deck. When a card has been used, the head pointer increases to move to the next position. If the end of the deck is reached, the head pointer returns to 0 and starts again.

The procedure `pickDeck`:

- takes the current player as a parameter
- outputs the text to be displayed from the first card in the queue
- adds or subtracts the amount to/from the current player's money
- increases the head pointer

Write, using pseudocode, the procedure `pickDeck`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(d) The procedure `checkAnimal`:

- Takes the current player as a parameter
- Accesses the data for the animal at the player's position in the array `board`
- If the animal is free, asks the player if they would like to purchase the animal and outputs its `name` and `cost`, if they choose to buy the animal, it calls the procedure `purchase()` with the player and animal as parameters
- If that player owns the animal, and it is not at level 3, it asks if they would like to upgrade the animal
- If they would like to upgrade, it calls the method `upgrade` for that animal with the current player as a parameter
- If a different player owns the animal, it calls the method `getAmountToCharge()` for that animal, sending this value and the current player as parameters to the procedure `chargeStay()`

Write, using pseudocode, the procedure `checkAnimal`. **[10]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

END OF QUESTION PAPER

OCR
Oxford Cambridge and RSA

Practice Paper 2

GCE Computer Science

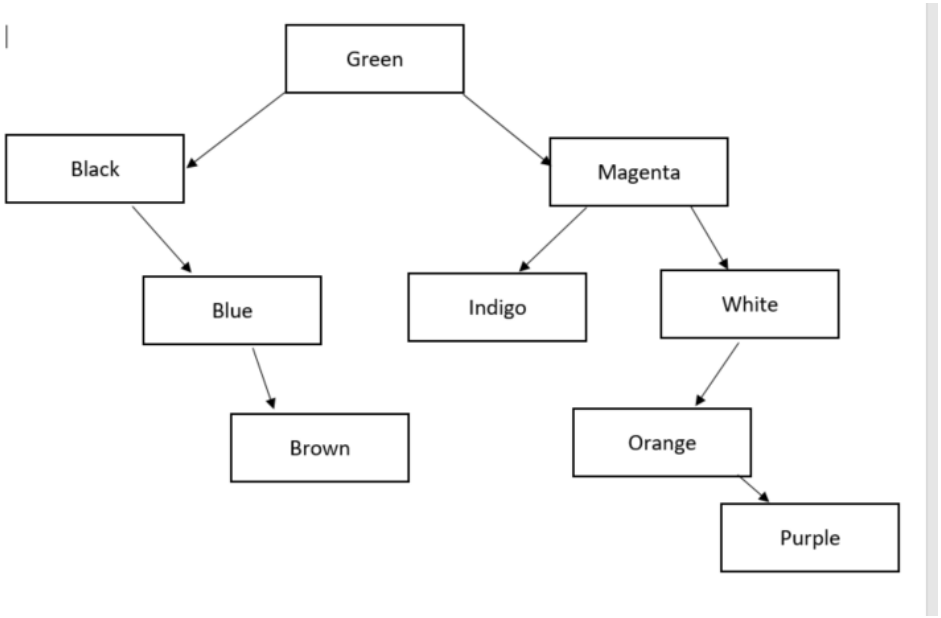
H446/02 Algorithms and Programming

Duration 2 hours 30 minutes

MAXIMUM MARK 140

FINAL

This document consists of 27 pages

Question			Answer	Marks	Guidance
1	a	i	1 mark per bullet to max 3 • A data structure • Consists of nodes • That have sub nodes (children) • First node is called the root • Lines that join nodes are called branches	**3** **AO1.1** **(3)**	
1	a	ii	1 mark per bullet to Max 1 • In a binary search tree, each node only has max. 2 sub nodes • If a child node is less than a parent node, it goes to the left of the parent. • If a child node is greater than a parent node, it goes to the right of the parent.	**1** **AO1.2** **(1)**	
1	b		1 mark for each node as a correct sub-node  	**4** **AO2.1** **(4)**	Allow follow through e.g. if white is incorrect, but orange follows through in a logical position.

Question			Answer	Marks	Guidance
1	c	i	2 marks for correct order • 5, 31, 20, 48, 45, 60 • 92, 88, 98, 76, 50 1 mark per bullet to max 3 for explanation • Visit all nodes to the left of the root node • Visit right • Visit root node • Repeat three points for each node visited	**5** **AO1.1 (1)** **AO1.2 (2)** **AO2.1 (2)**	
1	c	ii	2 marks for correct order • 50, 45, 76 • 20, 48, 60, 98, 5, 31, 88, 92 1 mark per bullet to max 3 for explanation • Visit root node • Visit all direct subnodes (children) • Visit all subnodes of first subnode • Repeat three points for each subnode visited	**5** **AO1.1 (1)** **AO1.2 (2)** **AO2.1 (2)**	
1	d	i	1 mark for the correct pointers for nodes 1, 2 and 3	**3** **AO2.1 (3)**	Ignore any change to FP, additional nodes and Right Pointers to node 4 and 5

Question	Answer	Marks	Guidance

Array Index	Left Pointer	Data	Right Pointer
0	1	68	2
1	3	30	
2	6	73	5
3	4	22	
4		1	
5		90	
6		70	

<table>
<tr><td>1</td><td>d</td><td>ii</td><td>To identify where the next element will be placed</td><td>1
AO1.2 (1)</td><td></td></tr>
<tr><td>1</td><td>d</td><td>iii</td><td>1 mark per bullet
• Adding data 6 and 100
• …In the correct order
• Updating the Right Pointers of nodes 4 and 5
• Updating the Free Pointer to 9
FP: 9
<table><tr><th>Array Index</th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr><tr><td>0</td><td>1</td><td>68</td><td>2</td></tr><tr><td>1</td><td>3</td><td>30</td><td></td></tr><tr><td>2</td><td>6</td><td>73</td><td>5</td></tr><tr><td>3</td><td>4</td><td>22</td><td></td></tr><tr><td>4</td><td></td><td>1</td><td>7</td></tr><tr><td>5</td><td></td><td>90</td><td>8</td></tr><tr><td>6</td><td></td><td>70</td><td></td></tr><tr><td>7</td><td></td><td>6</td><td></td></tr><tr><td>8</td><td></td><td>100</td><td></td></tr></table></td><td>4
AO2.1 (4)</td><td>Allow follow through for errors from 1di</td></tr>
<tr><td>2</td><td>a</td><td></td><td>1 mark per bullet
• Weighted/Undirected
• Graph</td><td>2
AO1.2 (2)</td><td></td></tr>
<tr><td>2</td><td>b</td><td>i</td><td>E.g. Weighting/cost based on estimated distance from final node</td><td>1
AO2.1</td><td></td></tr>
<tr><td>2</td><td>b</td><td>ii</td><td>Used to speed up process of finding solution</td><td>1
AO1.1</td><td></td></tr>
</table>

<table>
<tr>
<td>2</td>
<td>b</td>
<td>iii</td>
<td>1 mark per bullet, max 7 for calculations/explanation, max 1 for correct final path

• Visiting H with correct heuristic
• Visiting G and N from H
o Calculating correct distance+heuristic for G and N
• Identifying G as the smallest value
• Visiting L and M from G
o Calculating distance+heuristic for L and M
• Identifying L as the smallest value
• Visiting E
o Calculating distance+heuristic for E

• Final path: H-G-L-E

e.g.
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Heuristic</th><th>distance+heuristic</th><th>previous node</th></tr>
<tr><td>H</td><td>0</td><td>80</td><td>80</td><td>-</td></tr>
<tr><td>G</td><td>25</td><td>70</td><td>95</td><td>H</td></tr>
<tr><td>N</td><td>210</td><td>90</td><td>300</td><td>H</td></tr>
<tr><td>L</td><td>51+25=76</td><td>50</td><td>126</td><td>G</td></tr>
<tr><td>M</td><td>176+25=201
<s>233+210=443</s></td><td>20</td><td>221
<s>463</s></td><td>G
<s>N</s></td></tr>
<tr><td>E</td><td>307+76=383</td><td>0</td><td>383</td><td>L</td></tr>
</table>
</td>
<td>8
AO1.2
(4)
AO2.1
(4)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>iv</td>
<td>1 mark for decision, 2 marks for effect
e.g.
Decision:
• Choosing which node to take next
• The shortest distance+heuristic is taken

Effect:
• All adjoining nodes from this new node are taken
• Other nodes are compared again in future checks
• Assumed that this node is a shorter distance</td>
<td>3
AO2.1
(3)</td>
<td></td>
</tr>
</table>

			• Adjoining nodes may not be shortest path … • ….may need to backtrack to previous nodes		
2	**c**		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of concurrent processing; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided (searching algorithms). Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistenly relevant and well considered. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of concurrent processing; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. (searching algorithms) Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of concurrent processing with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** • Carrying out more than one task at a time • Multiple processors • Each processor performs simultaneously • Each processor performs tasks independently and/or • A program has multiple threads • Each thread starts and ends at different times • Each thread overlaps • Each thread runs independently **AO2: Application** • Each processor/thread performs a search in a different direction • Rather than going down one path, go down 2+ • E.g. apply different searches simultaneously - perform breadth-first and depth-first simultaneously • E.g. A* take the two shortest routes at each decision point, update same table • Linear search can have multiple processors searching different areas at the same time. • Binary search doesn't benefit from an increase in speed with additional processors

			knowledge and understanding to the context provided (searching algorithms). The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		**AO3: Evaluation** Candidates will need to evaluate the benefits and drawbacks of concurrent processing in searching e.g. • Possibly find solution faster • Takes up more memory • Increase program throughput • May waste time investigating inefficient solutions • More difficult to program especially to cooperate • More memory intensive • Linear search scales very with additional processors • Binary search can perform better on large data sets with one processor than linear search with many processors
3	**a**		1 mark for each feature e.g. • Involves calculations • Has inputs, processes and outputs • Involves logical reasoning	**2** **AO1.2** **(2)**	Allow any suitable feature
3	**b**		**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of decomposition and abstraction; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate provides a thorough discussion which is well balanced. Evaluative comments are consistenly relevant and well considered *There is a well-developed line of reasoning which is clear and logically*	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3** **(3)**	**AO1: Knowledge and Understanding** **Indicative content** Decomposition: • splits problem into sub-problems • splits these problems further • until each problem can be solved • Allows the use of divide and conquer Abstraction

			structured. The information presented is relevant and substantiated. **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of decomposition and abstraction; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of decomposition and abstraction with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.		• Removing unnecessary elements using symbols • Removing unnecessary design/programming/ computational resources **AO2: Application** • Split the simulation into subparts • E.g. generating rooms, patients, people, scenarios, interaction • E.g. replacing how instruments look with shapes, minimise features of human body **AO3: Evaluation** e.g. • Increase speed of production • Assign areas to specialities • Allows use of pre-existing modules • Allows re-use of new modules • Need to ensure subprograms can interact correctly • Can introduce errors • Reduces processing/memory requirements • Increases response speeds of programs
3	**c**		2 marks for definition, max 2 for application Caching: • Data that has been used is stored in cache/ram in case it is needed again • Allows faster access for future use	**4** **AO1.1** **(2)** **AO2.1** **(2)**	Allow any reasonable example A well-developed example can gain two marks

			Application: e.g. • Store patients' details/conditions/appearance • Design of people in the simulation • Design of specific rooms		
3	**d**	**i**	1 mark per bullet • How the times scales as data size increases • O(n) = linear complexity • Increases at the same rate as the number of data items increases	**3** **AO1.1** **(1)** **AO2.1** **(2)**	

<table>
<tr><td>3</td><td>d</td><td>ii</td><td>1 mark each
A = exponential
B = logarithmic</td><td>2
AO1.1
(2)</td><td></td></tr>
<tr><td>3</td><td>d</td><td>iii</td><td>1 mark for recommendation, max 3 for explanation
• Recommend: Solution B
Justification
• A’s space does not scale well when increased in number of items
• B’s space scales well / does not increase significantly with number of items
• As n increases at some point a will require significantly more space than B
• Both have same time complexity so need to look at space</td><td>4
AO1.2
(2)
AO2.1
(2)</td><td></td></tr>
<tr><td>3</td><td>e</td><td></td><td>1 mark per bullet, max 2 for each tools
Breakpoints
• Use to test the program works up to/at specific points
• Check variable contents at specific points
• Can set a point where the program stops running

Stepping
• Can set the program to run line by line
• Slow down/watch execution
• Find the point where an error occurs</td><td>4
AO1.1
(2)
AO1.2
(2)</td><td></td></tr>
<tr><td>4</td><td>a</td><td></td><td>1 mark for each correct swap identified/described

<table>
<tr><td>sheep</td><td>rabbit</td><td>dog</td><td>fox</td><td>cow</td><td>horse</td><td>cat</td><td>deer</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>cow</td><td>horse</td><td>cat</td><td>deer</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>horse</td><td>cow</td><td>cat</td><td>deer</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>fox</td><td>dog</td><td>horse</td><td>cow</td><td>deer</td><td>cat</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>fox</td><td>horse</td><td>dog</td><td>cow</td><td>deer</td><td>cat</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>fox</td><td>horse</td><td>dog</td><td>deer</td><td>cow</td><td>cat</td></tr>
<tr><td>sheep</td><td>rabbit</td><td>horse</td><td>fox</td><td>dog</td><td>deer</td><td>cow</td><td>cat</td></tr>
</table></td><td>6
AO1.2
(3)
AO2.1
(3)</td><td></td></tr>
<tr><td>4</td><td>b</td><td></td><td>1 mark per bullet to max 7
• Correct function declaration, taking string1 and string2 as parameters
• Use of a flag</td><td>7
AO2.1
(1)
AO2.2</td><td>Allow reversed true and false</td></tr>
</table>

			• Looping through string2 by some means
			• Using string manipulators to check either letters or substrings of string2
			• Correctly setting return value to `true`
			• Returning `true` or `false` accordingly
			• Comments that explain how the algorithm works
			
			e.g.
			`function contains(string1, string2)`
			`  wordInside = false`
			`  for i = 0 to (string2.length – string1.length)`
			`  if string2.substring(i, string1.length) == string1 then`
			`  wordInside=true`
			`  endif`
			`  next i`
			`  return wordInside`
			`endfunction`
				(2)
				AO3.2
				(4)	

4	**c**	**i**	1 mark per bullet to max 4 • Merge sort splits the data • Merge sorts the split data as it is put back together • Bubble moves through the data in a linear way • Bubble moves through the data repeatedly • Merge is more efficient with larger volumes of data to sort • Merge may require more memory space	**4** **AO1.1** **(1)** **AO1.2** **(2)** **AO2.1** **(1)**	Allow points by demonstration/example
4	**c**	**ii**	1 mark per example e.g. • Insertion • Quick	**2** **AO1.1** **(2)**	
4	**d**		1 mark for each bullet • Duck is smaller than goat • Duck is less than frog/elephant • Duck is equal to duck/less than elephant so only duck left	**3** **AO2.1** **(3)**	

5	**a**	**i**	1 mark per bullet to max 3
			• Declaring the procedure and taking a player ID as parameter
			• Setting `playerID` to parameter
			• Setting `boardPosition` to 0 and money to 2000
			
			e.g.
			`public procedure new(thePlayerID)`
			`  playerID = thePlayerID`
			`  boardPosition = 0`
			`  money = 2000`
			`endprocedure`
				3
				AO2.2
				(2)
				AO3.2
				(1)	

5	**a**	**ii**	1 mark per bullet to max 4
			• Declaration as public and procedure, named constructor/new and Taking all correct parameters (missing currentLevel)
			• Sets currentLevel to 0
			• Setting all the data…
			• …to the matching parameters taken
			e.g.
			`public procedure new(theName, theCost, theL0, theL1,`
			`theL2, theL3, theImageLink, theSetSquare, theOwned)`
			`  name = theName`
			`  currentLevel = 0`
			`  cost = theCost`
			`  L0 = theL0`
			`  L1 = theL1`
			`  L2 = theL2`
			`  L3 = theL3`
			`  imageLink = theImageLink`
			`  setSquare = theSetSquare`
			`  owned = theOwned`
			`endprocedure`
				4
				AO2.2
				(2)
				AO3.2
				(2)	

5	**a**	**iii**	1 mark per bullet to max 2 • Creating new instance e.g. squirrel = new Animal • Parameters matching part (b)(i) e.g.	**2** **AO2.1** **(2)**	

			squirrel = new Animal(“Squirrel”, 1000, 10, 50, 100, 500, “squirrel.bmp”, 6, “free”)		

5	**b**	**i**	1 mark for each correctly completed space
			
			function playerMove(currentPlayer)
			dice1 = random(1,6)
			dice2 = random(1,6)
			position = **currentPlayer.getPosition()** + dice1 + dice2
			if **dice1** == dice2 then
			pickDeck(currentPlayer)
			endif
			
			if position > 25 then
			currentPlayer.setMoney(currentPlayer.getMoney() + **500)**
			position = position - **26**
			endif
			if position == **13** then
			missAGo(currentPlayer)
			elseif position != 0 then
			checkAnimal(currentPlayer)
			endif
			return position
			endfunction
				6
				AO2.2
				(3)
				AO3.2
				(3)	

5	**b**	**ii**	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of passing values by reference and by value; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate	**9** **AO1.1** **(2)** **AO1.2** **(2)** **AO2.1** **(2)** **AO3.3**	**AO1: Knowledge and Understanding** **Indicative content** By Value • sends the actual value • if changes are made then only the local copy is amended By Reference • sends a pointer to the value

			provides a thorough discussion which is well balanced. Evaluative comments are consistently relevant and well considered *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of passing values by reference and by value; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of passing values by reference and by value with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.	**(3)**	• The actual value is not sent/received • If changed the original is also changed when the subroutine ends **AO2: Application** • Send by value • The `currentPlayer` value is not /does not need to be changed in the subprogram • Send by reference • The `currentPlayer` value is updated **AO3: Evaluation** • ByValue creates new memory space… • ByReference means existing memory space is used • Depends if original variable is local/global • If local and just referenced, send by value • If original value needs editing send by reference • If passing by reference then instead of returning `position` the code could just amend `currentPlayer.position` • If passing by value there could be inconsistencies when `currentPlayer` is passed to other methods, for example `pickDeck`

<table>
<tr>
<td>5</td>
<td>c</td>
<td></td>
<td>1 mark per bullet to max 6
• Procedure declaration with correct name and parameter
• Outputting the correct text from deck at <code>headPointer</code>
• Sending to <code>currentPlayer.setMoney</code> …
• <code>getMoney</code> + <code>deck</code> at head pointer amount
• Increase the head pointer
• Set <code>headPointer</code> to 0 if position 40 or greater

<pre>
procedure pickDeck(currentPlayer)
     output(deck[headPointer].getTextToDisplay())
     amount = deck[headPointer].getMoney()
     currentPlayer.setMoney(currentPlayer.getMoney() +
amount)
      headPointer = headPointer + 1
      if headPointer == 40 then
          headPointer = 0
      endif
endprocedure
</pre>
</td>
<td>6
AO2.2 (4)
AO3.2 (2)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>d</td>
<td></td>
<td>• Declaring the procedure with correct parameters
• Check if the space/animal is free
o If free, outputting name and cost
o Checking if they want to buy
▪ Calling purchase with current player and animal
• If they own the animal, checking if they can upgrade
▪ If they can, asking if they want to upgrade outputting the cost
▪ If they want to, calling the <code>upgrade</code> method
• If they don’t own the animal
o Calling <code>chargeStay</code> with the amount to charge and the current player</td>
<td>10
AO2.2(5)
AO3.2(5)</td>
<td>Allow follow through for incorrect accessing of methods</td>
</tr>
</table>

e.g.

```
procedure checkAnimal(currentPlayer)

   if board[currentPlayer.getPosition()].owned == “free”
      answer = input(“Would you like to purchase “,
                board[position].getName(), “? It costs “,
                board[position].getCost()

      if answer = “yes” then
         purchase(currentPlayer, board[position])
      endif

   elseif board[currentPlayer.getPosition()].getOwned() == currentPlayer

       if board[currentPlayer.getPosition()].getCurrentLevel()
!= “L3”
          answer = input(“Upgrade? It costs “,
board[position].getCost())

           if answer == “yes” then
              board[currentPlayer.getPosition()].upgrade(currentPlayer)
           endif

        endif

   else
      amount = board[position].getAmountToCharge()
      chargeStay(amount, currentPlayer)
   endif
endprocedure
```